

AD-A184 261

COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT
USER'S GUIDE: A THREE (U) ARMY ENGINEER WATERWAYS
EXPERIMENT STATION VICKSBURG MS INFOR. F T TRACY

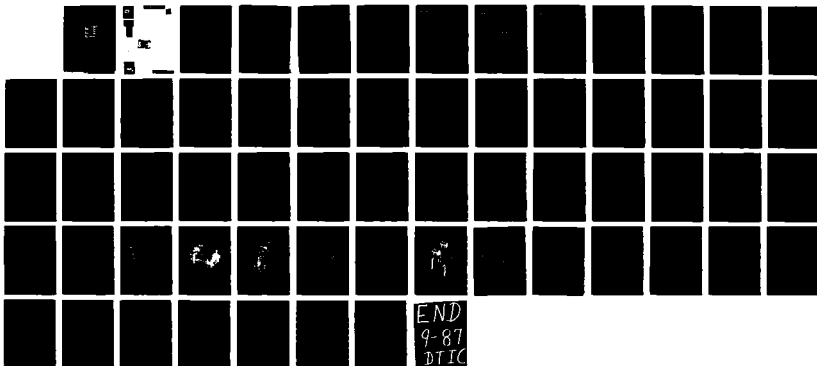
1/1

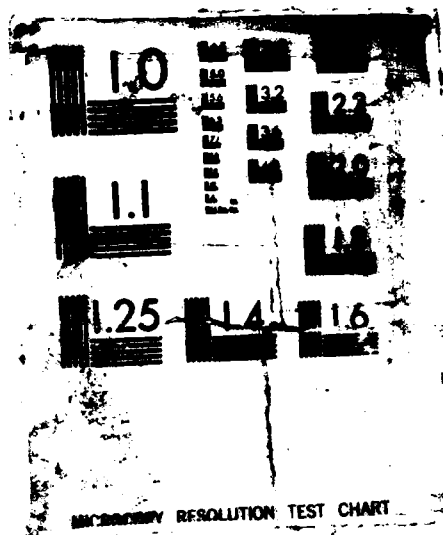
UNCLASSIFIED

JUN 87 WES/IR/ITL-87-3-1

F/G 12/5

NL





**Destroy this report when no longer needed. Do not return it
to the originator.**

**The findings in this report are not to be construed as an official
Department of the Army position unless so designated by other
authorized documents.**

**This program is furnished by the Government and is accepted and used by
the recipient with the express understanding that the United States
Government makes no warranties, expressed or implied, concerning the
accuracy, completeness, reliability, usability, or suitability for any par-
ticular purpose of the information and data contained in this program or
furnished in connection therewith, and the United States shall be under no
liability whatsoever to any person by reason of any use made thereof. The
program belongs to the Government. Therefore, the recipient further
agrees not to assert any proprietary rights therein or to represent this
program to anyone as other than a Government program.**

**The contents of this report are not to be used for
advertising, publication, or promotional purposes.
Citation of trade names does not constitute an
official endorsement or approval of the use of such
commercial products.**

PROGRAM INFORMATION

Description of Program

3DSAD, called X8100 in the Con conversationally Oriented Real-Time Program-Generating System (CORPS) Library, is a computer program for a three-dimensional stability analysis/design of hydraulic structures.

Coding and Data Format

3DSAD is written in FORTRAN and is operational on the following systems:

- a. US Army Engineer Waterways Experiment Station (WES), Vicksburg, Miss., and Division office Honeywell DPS/8.
- b. District office Harris 500.
- c. Cybernet Computer Service's CDC CYBER 175.
- d. Apollo microcomputer workstation.

Data must be in a prepared data file with line numbers or given interactively without line numbers. Output comes directly back to the terminal or micro-computer monitor. The terminal must be a Tektronix 4014 if graphics display is wanted.

How to Use 3DSAD

Directions for accessing the program on each of the three systems is provided below. It is assumed that the user can sign on the appropriate system before attempting to use 3DSAD. In the example initiation of execution commands below, all user responses are underlined, and each should be followed by a carriage return.

Honeywell System

After the user has signed on the system, the system command FORT brings the user to the level to execute the program. Next, the user issues the run command

RUN WESLIB/CORPS/X8100,R

to initiate execution of the program. The program is then run as described in this user's guide. A data file is typically prepared prior to issuing the run command. An example initiation of execution is as follows:

HIS TIMESHARING ON 03/04/81 AT 13.301 CHANNEL 5647
USER ID - R0KACASECON
PASSWORD - WHERE ARE YOU?
*FORT
*RUN WESLIB/CORPS/X8100,R

CYBERNET System

The log-on procedure is followed by a call to the CORPS procedure file

OLD,CORPS/UN=CECELB

to access the CORPS library. The file name of the program is used in the command

BEGIN,,CORPS,X8100

to initiate execution of the program. An example is:

84/12/05. 16.41.00. AC2F5DA
EASTERN CYBERNET CENTER SN904 NOS 1.4/531.669/20AD
FAMILY: KOE
USER NAME: CEROXX
PASSWORD -
XXXXXXXXX
TERMINAL: 23, NAMIAF
RECOVER/CHARGE: CHARGE,CEROEGC, CEROXX
\$CHARGE
12.49.07. WARNING (various information messages may appear here)

11/29 FOR IMPORTANT INFO TYPE EXPLAIN,WARNING. (Various information messages
may appear here.)

OLD,CORPS/UN=CECELB
/BEGIN,,CORPS,X8100

Harris 500 System

The log-on procedure is followed by a call to the program executable file, with the user typing the asterisk and file description

*CORPS,X8100

to initiate execution of the program. An example is:

"ACOE-ABLESVILLE (H500 V3.1)"
ENTER SIGN-ON
1234ABC,STRUCT

**GOOD MORNING STRUCTURES, IT'S 7 DEC 84 08:33:12
AED HARRIS 500 OPERATING HOURS 0700-1800 M-S
*CORPS,X8100

Apollo Microcomputer Workstation

If 3DSAD is installed under the directory CORPS with the executable file name X8100, then type

/CORPS/X8100

How to Use CORPS

The CORPS system contains many other useful programs which may be catalogued from CORPS by use of the LIST command. The execute command for CORPS on the Honeywell system is:

RUN WESLIB/CORPS/CORPS,R
ENTER COMMAND (HELP,LIST,BRIEF,MESSAGE,EXECUTE, OR STOP)
*?LIST

on the Cybernet system, the commands are:

OLD,CORPS/UN=CECELB
BEGIN,,CORPS
ENTER COMMAND (HELP,LIST,BRIEF,MESSAGE,EXECUTE, OR STOP)
*?LIST

on the Harris system, the commands are:

*CORPS
ARE YOU USING A PRINTER TERMINAL OR CRT?
ENTER P OR C
P
CORPS SYSTEM COMMANDS:
BRIEF - LIST EXPLANATION OF A PROGRAM.
EXECUTE - RUN A CORPS PROGRAM.
LIST - LIST THE AVAILABLE CORPS PROGRAMS.
STOP - EXIT FROM CORPS SYSTEM MACRO.
HELP - HELP AND EXPLANATION OF CORPS
SYSTEM AND THE RUNNING OF ITS MACRO.

NOTE: COMMANDS MAY BE ABBREVIATED TO THE
FIRST LETTER OF THE COMMAND.

ENTER COMMAND (BRIEF,EXECUTE,LIST,HELP,STOP):
LIST

This capability is not yet implemented on the Apollo.

ELECTRONIC COMPUTER PROGRAM ABSTRACT

TITLE OF PROGRAM A Three-Dimensional Stability Analysis/ Design Program (3DSAD) (X8100)		PROGRAM NO. 713-F3-R008
PREPARING AGENCY US Army Engineer Waterways Experiment Station, Information Technology Laboratory, PO Box 631, Vicksburg, MS 39180-0631		
AUTHOR(S) Fred T. Tracy	DATE PROGRAM COMPLETED September 1986	STATUS OF PROGRAM
		PHASE OP STAGE

A. PURPOSE OF PROGRAM

This program does an overturning and sliding stability analysis for complex three-dimensional structures. General shapes and specific structure types can be handled. The general modules are (1) geometry--to define, display, and compute mass properties, (2) loads--to define, display and compute loads, (3) analysis--to perform a base analysis assuming a rigid body on an elastic foundation, (4) free-body--to use clipping capability for a free-body analysis.

B. PROGRAM SPECIFICATIONS

FORTRAN

C. METHODS

Geometry is defined by (1) two-dimensional cross sections swept or grown in the third dimension either linearly or axisymmetrically, (2) eight-node brick finite elements, and (3) clusters of planar faces and bicubic patches. Loads are computed by giving a direction to a volume or specifying a point load. Six standard load cases are provided for dams with design memorandum (DM) plate capability.

D. EQUIPMENT DETAILS

Graphics terminal with time-sharing system or stand-alone supermicrocomputer.

E. INPUT-OUTPUT

Input is in the form of an input file combined with an interactive session. Output is displayed on the terminal or sent to a remote drum plotter.

F. ADDITIONAL REMARKS

The program is available through the CORPS library for the Honeywell DPS-8, Cybernet CDC, and Apollo computers.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

AD-A18426

REPORT DOCUMENTATION PAGE				
1a. REPORT SECURITY CLASSIFICATION		1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY Unclassified		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Instruction Report ITL-87-3		5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION USAEWES Information Technology Laboratory		6b. OFFICE SYMBOL (If applicable) WESIM-RS		7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) PO Box 631 Vicksburg, MS 39180-0631		7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION US Army Corps of Engineers		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Washington, DC 20314-1000		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
				WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) User's Guide: A Three-Dimensional Stability Analysis/Design Program (3DSAD) Report 1 Revision 1: General Geometry Module				
12. PERSONAL AUTHOR(S) Tracy, Fred T.				
13a. TYPE OF REPORT Report 1 of a series		13b. TIME COVERED FROM Jun 85 TO Nov 86		14. DATE OF REPORT (Year, Month, Day) June 1987
15. PAGE COUNT 59				
16. SUPPLEMENTARY NOTATION This report was prepared under the Computer-Aided Structural Engineering (CASE) Project. A list of published CASE reports is printed on the inside of the back cover				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) See reverse	
FIELD	GROUP	SUB-GROUP		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) → This module of the three-dimensional (3-D) stability analysis/design program (3DSAD) enables the engineer to describe the geometry of the 3-D structure, to interactively plot the desired structure, and then to obtain volume, weight, and centroid information for the structure. The general ways in which geometry is described are (a) two-dimensional cross sections extended in the third dimension, (b) eight-node brick elements, and (c) clusters of planar polygonal and bicubic patches. Curved segments can be circular, elliptical, or quadratic when using blocks and cubic when using patches. Objects can be copied, translated, rotated, or reflected about an axis as well. An arbitrary clipping plane can also be passed through the geometry.				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL			22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

SECURITY CLASSIFICATION OF THIS PAGE

→

Geometry
Structural analysis
Structural design

SECURITY CLASSIFICATION OF THIS PAGE

PREFACE

This report documents the General Geometry Module of the three-dimensional stability analysis/design (3DSAD) program. The module was developed and this report was written at the US Army Engineer Waterways Experiment Station (WES), Vicksburg, Miss., in the Information Technology Laboratory (ITL), formerly the Automation Technology Center, by Mr. Fred T. Tracy. The work was sponsored through funds provided WES by the Engineering and Construction Directorate, Office, Chief of Engineers (OCE), US Army, under the Computer-Aided Structural Engineering (CASE) Project.

Specifications for the program were provided by the members of the CASE Task Group on 3-D Stability. The members of the task group during the initial period of development were as follows:

Mr. Charles Kling, Mobile District (Chairman)
Mr. Robert Haavisto, Sacramento District
Mr. John Hoffmeister, Nashville District
Mr. Gerrett Johnson, Seattle District
Mr. Thomas Mudd, St. Louis District
Mr. William Holtham, New England Division

Members of the task group during the latest development were:

Mr. Kling (Chairman)
Mr. Lavane Dempsey, St. Paul District
Mr. Jack Duckworth, Federal Energy Regulatory
Commission (FERC)
Mr. Steve Freitas, Sacramento District
Mr. Holtham
Mr. Johnson
Mr. Tom Leicht, St. Louis District

Mr. Donald R. Dressler, Structures Branch, Engineering and Construction Directorate, was the OCE point of contact. The work was done under the supervision of Dr. N. Radhakrishnan, Acting Chief (AC), ITL, and Mr. Paul K. Senter, AC, Information Research Division, ITL. Mr. Dressler and Dr. Radhakrishnan also contributed in the definition of general concepts for the development of 3DSAD. The report was prepared for publication by Ms. Gilda Miller, Editor, Information Products Division, ITL, WES.

COL Allen F. Grum, USA, was the previous Director of WES. COL Dwayne G. Lee, CE, is the present Commander and Director. Dr. Robert W. Whalin is Technical Director.



Dist

A-1

Avail and/or
Special

Codes

CONTENTS

	<u>Page</u>
PREFACE.....	1
PART I: OVERVIEW OF THE THREE-DIMENSIONAL STABILITY PROGRAM.....	3
The General Geometry Module.....	4
Coordinate System.....	5
Data Types.....	5
PART II: RUNNING THE PROGRAM.....	10
Commands.....	10
Practical Applications.....	33
REFERENCES.....	34
PLATES 1-3	
APPENDIX A: THEORY AND PROCEDURE FOR COMPUTING MASS PROPERTIES.....	A1
Introduction.....	A1
Modeling Technique.....	A1
Mass Properties--BLOCK.....	A1
Mass Properties--FACES.....	A6

USERS GUIDE: A THREE-DIMENSIONAL STABILITY

ANALYSIS/DESIGN PROGRAM (3DSAD)

REVISION 1: GENERAL GEOMETRY MODULE

PART I: OVERVIEW OF THE THREE-DIMENSIONAL STABILITY PROGRAM

1. The objective of the Computer-Aided Structural Engineering (CASE) Task Group on Three-Dimensional (3-D) Stability Analysis is to develop computer programs to aid design engineers perform stability computations for general 3-D structures (Tracy 1980; Tracy and Kling 1982; and Tracy, Kling, and Holtham 1983). To enable this, a computer program called 3DSAD (3-D stability analysis/design) has been developed in a modular fashion. Currently, 3DSAD has four "general" modules:

a. General Geometry Module.

- (1) Defines geometry based on two-dimensional (2-D) cross sections extended into the third dimension, eight-node brick elements, or clusters of planar polygonal or bicubic patches.
- (2) Performs centroid, volume, and weight computations on described geometry.
- (3) Employs interactive graphics extensively.

b. General Loads Module. This module uses "pressure volumes" and point loads to compute forces and moments for a general 3-D structure. It will eventually compute loads based on input of geometry, water levels, and soil strata descriptions only with the pressure volumes (or equivalent surface integration) generated automatically.

c. General Analysis Module. This module performs overturning, bearing, and sliding computations.

d. Free-Body Module. This module "clips" the structure and loads by an arbitrary plane to produce a "free-body" so that computations can be performed on the new part.

The engineer performing an analysis of any 3-D structure can interact directly with these modules.

2. In addition to the general capabilities that are useful for any 3-D structures, 3DSAD also provides for simplified geometry and load input along with criteria check modules for certain structures. This latter capability will permit interactive design of these structures. Examples of specific structures for which modules can be developed are dams, locks, walls, powerhouses, and pumping stations.

3. A specific structure input module requires less data than that for a general structure. Modules of this type will interact with the General Geometry Module and the General Loads Module to define the geometry and loads internally in the program. After analysis, a specific structure criteria check module will verify pertinent values, change dimensions (if necessary), and cycle through the computations. A general schematic of the 3DSAD program is shown in Figure 1.

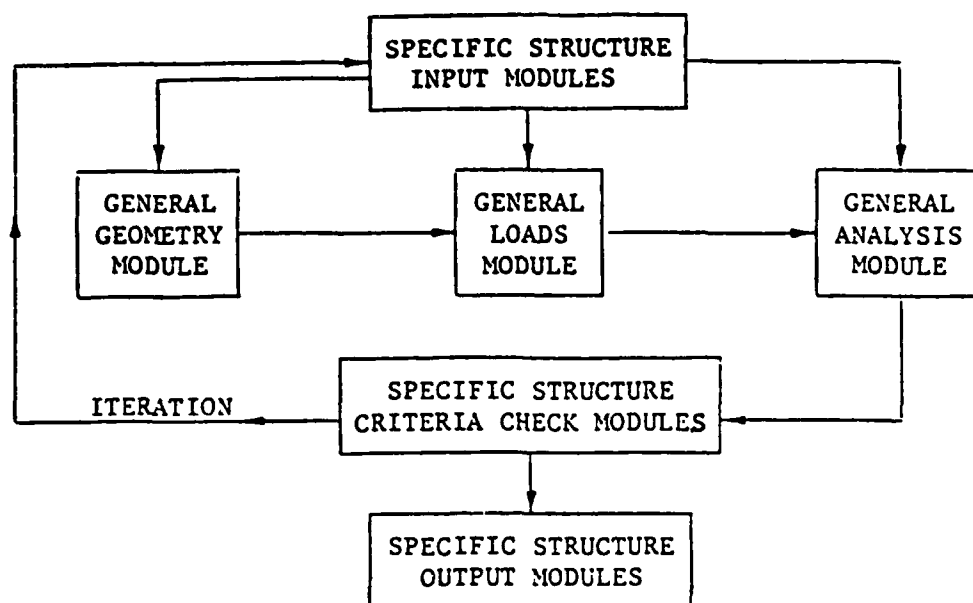


Figure 1. General schematic of 3DSAD

4. The 3DSAD program has been developed in phases. During the first phase, the first three general modules were developed. This approach enabled the stability analysis of any 3-D structure although the input is more complicated than need be for certain structures. In the subsequent phases, the special input and criteria check modules were developed for several specific structures. Also, new general modules such as the Free-Body Module have been added.

The General Geometry Module

5. This report is an updated user's guide for the General Geometry Module (the original is IR K-80-4). As stated in paragraph 1a this module enables the engineer to describe the geometry of a 3-D structure, to interactively plot the described structure, and then to obtain volume, weight, and

centroid information for the structure. In the stand-alone mode of operation for a general structure, the user first creates a data file and stores it in a permanent disc file. He then uses the graphics to verify the data. When satisfied, the user types

VOLUME

to obtain the resultant volume, weight, and centroid of the structure. Appendix A contains a mathematical presentation on how the mass properties are computed.

Coordinate System

6. The coordinate system used is shown in Figure 2. Note that X is to the right, Z is up, and Y is into the paper. This is a right-handed system.



Figure 2. Coordinate system

Data Types

7. Data are points, curves, surfaces, or solids. Three types of solid pieces can be used to describe the geometry. These are:

- a. Blocks
- b. Eight-node brick elements
- c. Clusters of surface patches to form a solid

Blocks

8. A block consists of a 2-D cross section defined in a plane and swept linearly or axisymmetrically to form a solid piece of geometry. Linearly swept cross sections can be defined in the X-Y, X-Z, or Y-Z planes which grow in the Z, Y, or X directions, respectively, to form the block. Figure 3 shows a typical cross section defined in the X-Z or default plane, and Figure 4 shows the generated block. A local polar coordinate system can also be established for defining the points and curves of both the cross section and the block. In this case, however, the sweep is axisymmetric instead of linear. Once created, a block can then be translated, rotated, copied, or reflected about a plane.

9. Figure 5 shows a data file and two cylindrical blocks: one

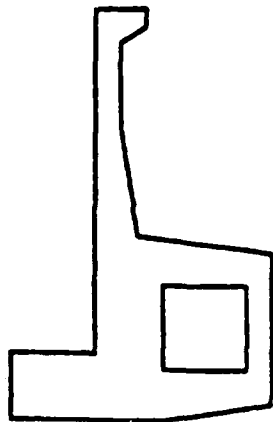


Figure 3. 2-D cross section

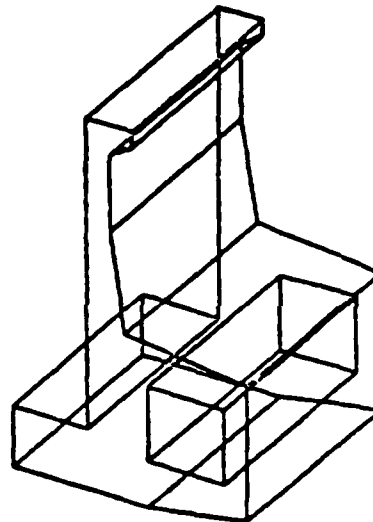


Figure 4. Generated block

```

10 XZ
20 POINTS 4
30 1 -5 0 0
40 2 0 0 5
50 3 5 0 0
60 4 0 0 -5
70 CIRCLE 1 2 5
80 CIRCLE 2 3 5
90 CIRCLE 3 4 5
100 CIRCLE 4 1 5
110 BLOCK BL1 .15 40
120 1 1
130 4 1 2 3 4
140 XY
150 POINTS 4
160 5 -5 20 5
170 6 0 15 5
180 7 5 20 5
190 8 0 25 5
200 CIRCLE 5 6 5
210 CIRCLE 6 7 5
220 CIRCLE 7 8 5
230 CIRCLE 8 5 5
240 BLOCK BL2 .15 40
250 1 1
260 4 5 6 7 8

```

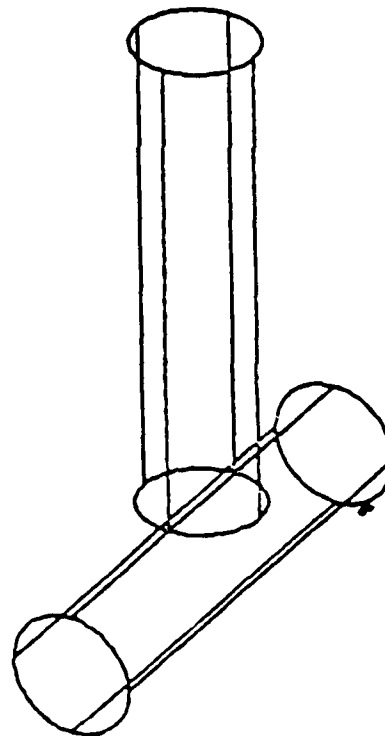


Figure 5. Two cylinders

generated from an X-Z cross section and the other from an X-Y cross section. Note that in the data file "XZ" indicated that the data refer to the X-Z plane and that "XY" is used when the switch is made to referring to the X-Y plane. Also, points are first defined, then any curves, and finally the block itself. The cross sections are specified by giving the point numbers defining the section in a counterclockwise positive manner according to the right-hand screw rule. For example, a cross section defined in the X-Z plane will grow into a block along the Y axis. Thus, if the user is looking in the negative Y direction, he must give the connectivity in a counterclockwise order to get a positive volume and weight. In like manner, a cross section defined in the X-Y plane grows in the Z direction and will have its connectivity specified in a counterclockwise manner by a user looking in the negative Z direction to obtain a positive volume. If the connectivity is reversed, the volume will be negative.

10. The line segments describing the cross section can be straight, circular, quadratic, or elliptical. Further, the section can grow smaller or larger as it is extended in the perpendicular direction. Any number of holes (for culverts, etc.) can be defined in the cross section as well.

Brick

11. The eight-node brick element is another useful way to describe geometry. Figure 6 shows a typical brick element.

Surface patches

12. In some cases, it is desirable to describe a solid piece of geometry by a group of surface patches. This program allows for planar faces with straight or cubic edges or bicubic patches (Figure 7).

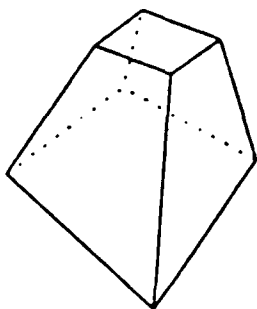


Figure 6. Brick element

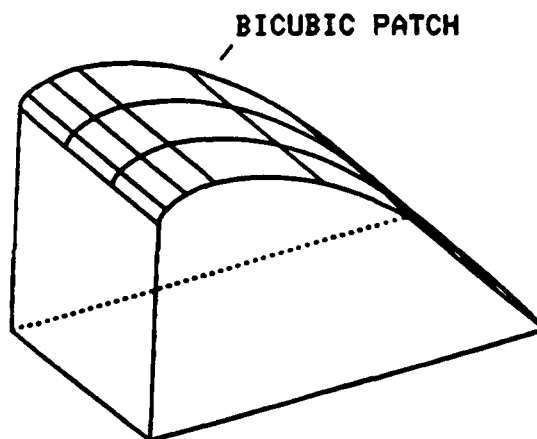


Figure 7. Bicubic patch

STRUCTURE TYPE OR GENERAL MODULE ?

13. If the user types a "?", the following message will appear:

A THREE-DIMENSIONAL STABILITY ANALYSIS/DESIGN PROGRAM
(3DSAD)

A PRODUCT OF

Computer Aided Structural Engineering

(CASE)

PROGRAM NO. 713-F3-RO-008

Computer Aided Design (CAD) OF STRUCTURAL STABILITY

() () () () () ()

ENTER ?, HELD, OR WHAT TO GET VALID RESPONSES.

ENTER STOP, END, QUIT, OR DONE TO TERMINATE PROGRAM.

STRUCTURE TYPE OR GENERAL MODULE ?

The user then gives

GEOM

for the General Geometry Module.

14. The next question the program asks is

RESTART FILE NAME OR CR?

when "CR" stands for carriage return. The restart file saves all data pertaining to building structure. These include any data input from another data file. The user gives a carriage return if he does not want a restart file.

15. The next question is

OUTPUT FILE NAME OR CR?

In this file is placed the resulting weight and centroid of the structure in the form of a point load; for example,

2 PTLD WT 17.584 20.000 1.004 0. 0. -8,067.939

The file consists of one line with line number 2, X centroid of 17.584, Y Centroid of 20.000, Z Centroid of 11.004, and weight of 8,067.939. A

carriage return is given if an output file is not wanted. The file is written when the RETURN command is given.

16. The third question is

COMMAND?

The commands are discussed in detail in the next section.

PART II: RUNNING THE PROGRAM

Commands

17. The program uses commands (PLOT, ROTP, etc.) to both build and plot the data. The commands are:

a. Data building:

XY XZ YZ RTZ
POINT
CIRCLE ELLIPSE
QUADRATIC
BLOCK FACE
BR8
TRANSLATE
ROTD COPY REFLECT

b. Utility:

INPUT VOLUME
END GO RETURN
CLIP CLEAR

c. Plotting:

PLOT WINDOW ZOOM
ROTP ISOMETRIC
LABEL NOLABEL
DASH HIDE SOLID
ERASE INITIALIZE

This list is obtained by typing "?" at the command level. Only the minimum number of letters of a command need to be given. The user can, however, type the entire word if he prefers. Commands and their accompanying data can be put into a data file or typed interactively while running the program. The basic command sequence is:

INPUT FILENAME	Read data from file FILENAME
PLOT	Plot data
VOLUME	Compute and print volume
END	End

18. Each command will now be described in detail. In giving the format for the commands, actual letters to be typed will be enclosed in the quotes to distinguish them from variable names. The quotes do not have to be typed when the user issues the command. The required letters are shown in all capitals;

the optional letters are shown in lower-case letters.

"XY"

19. The format for this command is

"XY"

This command turns on the flag that states that all "Circle" and "Ellipse" commands define circular and elliptical arcs in the X-Y plane, and all "Blocks" commands start with cross sections in the X-Y plane and grow in the Z direction (Figure 5 and the associated data file). This condition is held until an XZ, YZ, or "RTZ" command is encountered.

"XZ"

20. The format for this command is

"XZ"

This command is like XY except that circular and elliptical arcs are defined in the X-Z plane, and blocks start with cross sections in the X-Z plane and grow in the Y direction. XZ is assumed until an XY, YZ, or RTZ command is encountered. XZ is the default condition.

"YZ"

21. The format for this command is

"YZ"

This command is like XY and XZ except that circular and elliptical arcs are defined in the Y-Z plane, and blocks start with cross sections in the Y-Z plane and grow in the X direction. YZ remains in effect until an XY, XZ, or RTZ command is encountered.

"RTZ"

22. This command establishes a local polar coordinate system (Radius-Theta-Z) in space. The Z axis is established by two specified points (X1, Y1, Z1) and (X2, Y2, Z2), and (X1, Y1, Z1) is the new origin. The format of the command is

"RTZ" X1 Y1 Z1 X2 Y2 Z2

Theta is zero in the R-Theta plane where the new local Z axis strikes the R-Theta plane when projected vertically. If the new local Z axis is vertical, theta is zero along the line parallel to the X axis and passing through

(X1, Y1, Z1). All coordinates of points are specified in this system until another is specified. Also, the BLOCK command now expects an angle in degrees to produce a volume swept by a rotation rather than a translation. The cross section for a block can have curved or straight line segments as before, but the points defining the cross section must all have the same THETA value. Figure 8 shows a sample data file with its plot. RTZ is in effect until the XY, XZ, or YZ command is given.

```

10 RTZ 6. 5. 0. 6. 5. 1.
20 POINTS 4
30 1 10. 30. 2.
40 2 10. 30. 10.
50 3 15. 30. 10.
60 4 15. 30. 2.
70 BLOCK BL1 .1 90.
80 1. 1.
90 4 1 2 3 4

```

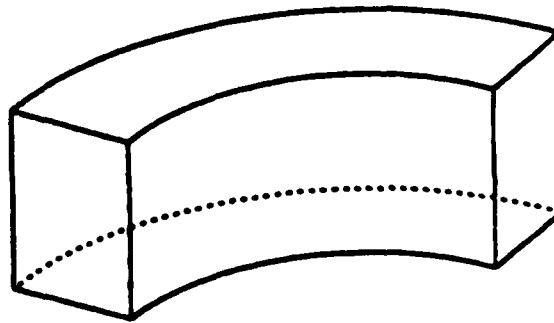


Figure 8. RTZ example

"Points"

23. The user first defines points using the POINTS command. The format is

"Points" NPT

where NPT is the number of points. After this line, an identification number and (X, Y, Z) coordinates for each point are given.

24. Interactive mode. An example of the POINTS command when made interactively on the CDC system is shown:

```

CINNABD ?
POINTS 14
N, X, Y, Z
1 0 0 565
2 44 0 565
3 44 0 577
4 40 0 577
5 40 0 597
6 22 0 617
7 22 0 633
8 4 0 633
9 4 0 577
10 0 0 577
11 9 0 585
12 9 0 595
13 19 0 595
14 19 0 585
COMMAND ?
?
```

When RTZ has been specified, X, Y, and Z become R, THETA, and Z in the local polar coordinate system.

25. Data file mode. The same data are put in a file as shown below:

```

10 POINTS 14
20 1 0 0 565
30 2 44 0 565
40 3 44 0 577
50 4 40 0 577
60 5 40 0 577
70 6 22 0 617
80 7 22 0 633
90 8 4 0 633
100 9 4 0 577
110 10 0 0 577
120 11 9 0 585
130 12 9 0 595
140 13 19 0 595
150 14 19 0 585
```

"Circle"

26. After the user has defined points to work with, he must then define any curved line segments. That is, line segments between points are assumed straight unless otherwise specified. The possible ways of using Circle command are

```

"Circle" N1 N2 R
"Circle" N1 N2 R "Left"
"Circle" N1 N2 R "Right"
```

N1 and N2 are two point numbers connected by a circular arc of radius R.

"Left" or "Right" designates to which side of the line segment N1 to N2 is the center of the circle. The question of left or right depends, of course, on which side of the cross section you are looking. Therefore, the user must be looking in the counterclockwise positive direction to correctly decide if the center of the circle is left or right of the line segment. The following data file results in the plot shown in Figure 9:

```

10 XZ
20 POIN 4
30 1 0. 0. -5.
40 2 5. 0. 0.
50 3 0. 0. 5.
60 4 -5. 0. 0.
70 CIRC 2 1 5.
80 CIRC 3 2 5. LEFT
90 CIRC 3 4 5. RIGHT
100 CIRC 1 4 5.

```

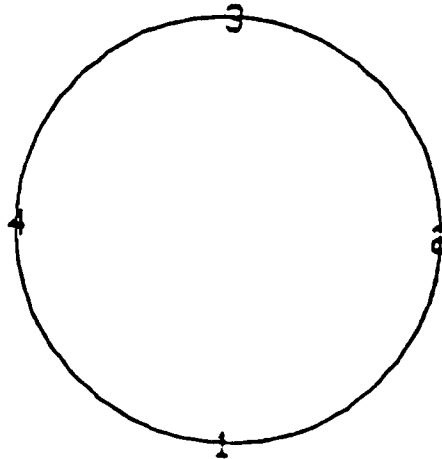


Figure 9. "Circle" command

27. The circle will be defined in the X-Y, X-Z, Y-Z, or R-Z plane, depending on whether or not the XY, XZ, YZ or RTZ command has been given before the circle is defined. XZ is the default option.

"ELlipse"

28. The user can also define an elliptical line segment. The different uses of the "ELlipse" command are

```

"ELlipse" N1 N2 A B
"ELlipse" N1 N2 A B "Left"
"ELlipse" N1 N2 A B "Right"

```

N1 and N2 are two point numbers connected by an elliptical arc having semi-major axis length A and semiminor axis length B. "Left" and "Right" have the same meaning as in the "Circle" command. The following file results in the plot shown in Figure 10.

```

10 XZ
20 POIN 4
30 1 0. 0. -5.
40 2 10. 0. 0.
50 3 0. 0. 5.
60 4 -10. 0. 0.
70 ELLI 2 1 10. 5.
8Q ELLI 3 2 10. 5. LEFT
90 ELLI 3 4 10. 5. RIGHT
100 ELLI 1 4 10. 5.

```

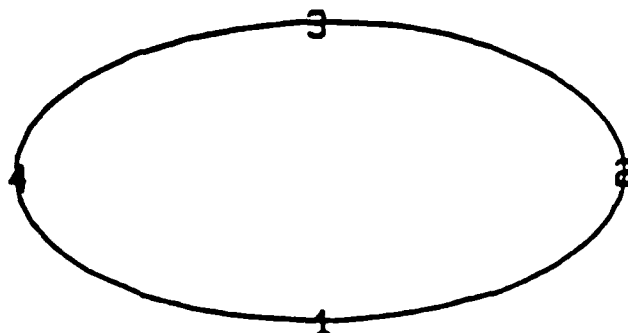


Figure 10. ELLipse command

Note that the semimajor and semiminor axes are always parallel to the co-ordinate axes.

29. As with the "Circle" command, the "ELLipse" command will be defined in one of the principal planes, depending on whether or not XY, XZ, YZ, or RTZ has been previously given.

"Quadratic"

30. The quadratic line is provided for cases when the user needs a curved line segment which is not circular or elliptical. The command format is

"Quadratic" N1 N2, XQQ, YQQ, ZQQ

N1 and N2 are the point numbers that the quadratic line segment is drawn between and (XQQ, YQQ, ZQQ) is an interpolation point (Figure 11) that the curve must go through. The following data file results in the plot shown in Figure 12:

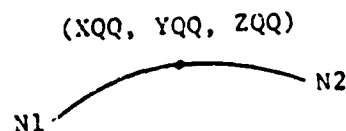


Figure 11. Quadratic plotting

```

10 POIN 4
20 1 0 0 -5
30 2 5 0 0
40 3 0 0 5
50 4 -5 0 0
60 QUAD 1 2 4 0 -3
70 QUAD 2 3 3 0 4
80 QUAD 3 4 -4 0 3
90 QUAD 4 1 -3 0 -4

```

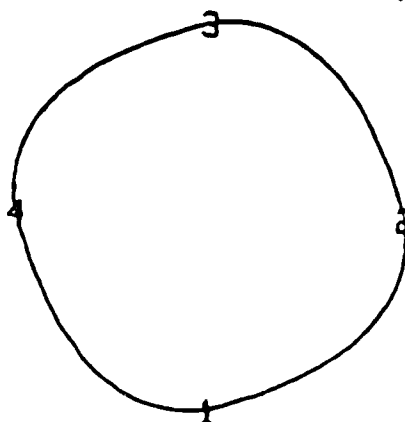


Figure 12. "Quadratic" command

when RTZ has been specified, XQQ, YQQ, and ZQQ become RQQ, TQQ, and ZQQ in the local polar coordinate system.

"BBlock"

31. Perhaps the most useful way to define solids is the BLOCK command as provided in this program. A 2-D cross section defined in one of the principal planes (X-Y, X-Z, Y-Z or local R-Z) is allowed to grow in the third (perpendicular) direction. The format for this command is

"BBlock" NAME DENS DEPTH NHOLE

where NAME is a four-character name of the block, DENS is the density, DEPTH is how far the cross section is extended, and NHOLE is the number of holes in the cross section. After the "BBlock" command is given in the interactive mode, the following questions are asked for the outer boundary and each hole:

SFX, SFZ, XAPEX, ZAPEX, HFX, HFZ ?

CONNECTIVITY DATA ?

If XY has been typed previously, SFZ is replaced by SFY, HFZ by HFX, and ZAPEX by YAPEX. In like manner, YZ typed previously results in replacing SFX by SFY, HFX by HFY, and XAPEX by YAPEX. Although the following discussion uses the default X-Z plane, the principles are the same for XY and YZ.

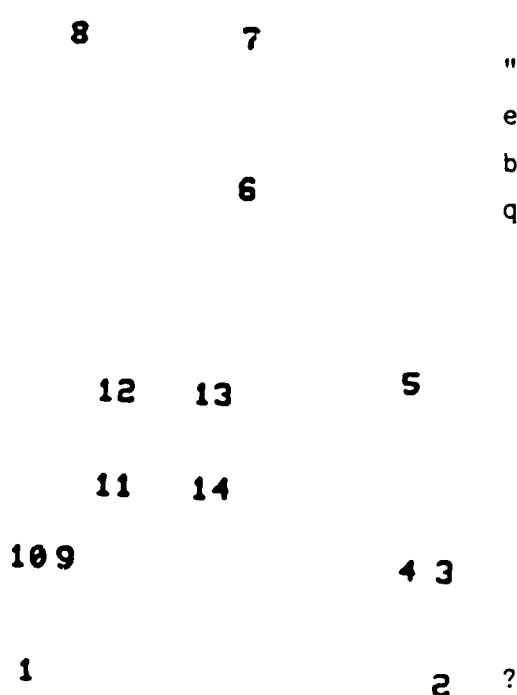
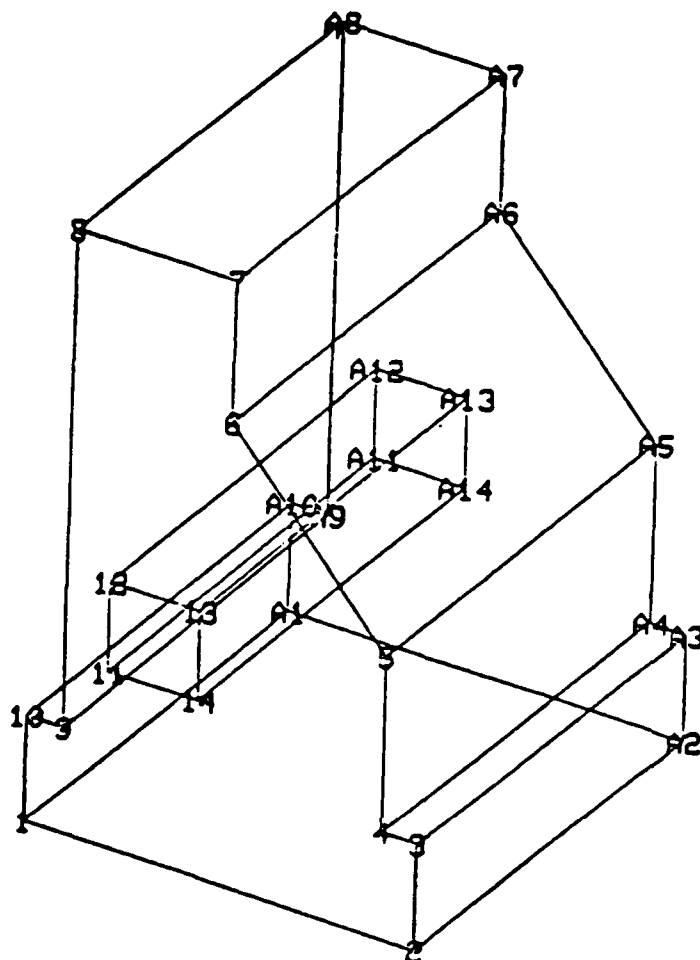


Figure 13. 14 Points

32. To understand these questions and the "BBlock" command, first consider the following example. The 14 points in Figure 13 have just been read in by the "POints" command. The question and answer sequence is typed.

```
COMMAND ?
BLOCK BL1 .1 50. 1
OUTER BOUNDARY DATA
SFX, SFZ, XAPEX, ZAPEX, HFX, HFZ ?
1. 1.
CONNECTIVITY DATA ?
10 10 9 8 7 6 5 4 3 2 1
DATA FOR HOLE 1
SFX, SFZ, XAPEX, ZAPEX, HFX, HFZ ?
1. 1.
CONNECTIVITY DATA ?
4 14 13 12 11
COMMAND ?
```

Figure 14 shows the generated block. If these data were placed into a data file, it would appear as



```

160 BLOCK BL1 .1 50. 1
170 1. 1.
180 10 10 9 8 7 6 5 4 3 2 1
190 1. 1.
200 4 14 13 12 11

```

Figure 14. Generated block

Note that the number of points, followed by the numbers given in counter-clockwise order, form the connectivity data of the outer boundary. The connectivity data for each hole are the same except that the point numbers are given in clockwise order.

33. Each original point of the cross section (numbered 1, 2, 3, etc.) will have a corresponding point generated a distance DEPTH in the Y direction (A1, A2, A3, etc.). The new (X, Y, Z) coordinates of these points are computed by

$$\begin{aligned}
 Y_{NEW} &= Y_{OLD} + DEPTH \\
 X_{NEW} &= (X_{OLD} - X_{APEX}) * SFX + X_{APEX} \\
 Z_{NEW} &= (Z_{OLD} - Z_{APEX}) * SFZ + Z_{APEX}
 \end{aligned}$$

Note that if SFX and SFZ are equal to one, then

$$XNEW = XOLD$$

$$ZNEW = ZOLD$$

These are independent of the apex (XAPEX, ZAPEX). This allows XAPEX and ZAPEX in the data for Figure 14 to default to zero.

34. Figures 15 through 18 show some of the data files and their corresponding plots, which illustrate the impact of SFX, SFZ, XAPEX, and ZAPEX. SFX and SFZ are scale factors which scale the front X and Z coordinates to determine the X and Z coordinates for the back face. In like manner, HFX and HFZ determine the X and Z coordinates for points halfway between the front and back faces.

35. In all the previous examples, HFX and HFZ were zero. This generated straight lines connecting the front cross section with the back cross section. To obtain a quadratic variation (shown by the data file and plot in Figure 19), interpolation points are computed to a distance $DEPTH/2$ from the

```
10 POIN 4
20 1 0. 0. 0.
50 2 10. 0. 0.
40 3 10. 0. 10.
30 4 0. 0. 10.
60 BLOCK BL1 .1 15.
70 1. 1.
80 4 4 3 2 1
```

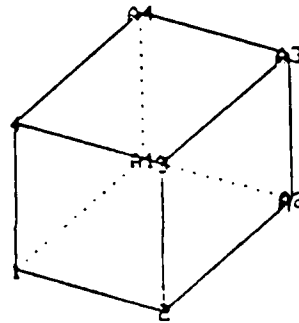


Figure 15. No scaling

```
10 POIN 4
20 1 0. 0. 0.
50 2 10. 0. 0.
40 3 10. 0. 10.
30 4 0. 0. 10.
60 BLOCK BL1 .1 15.
70 1. .5
80 4 4 3 2 1
```

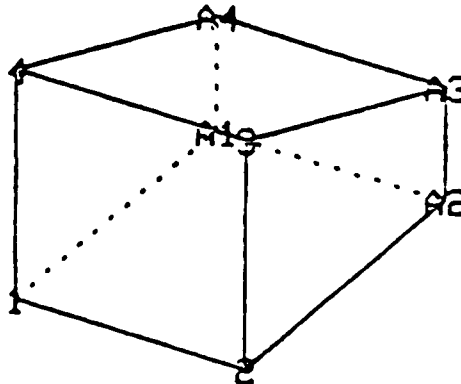


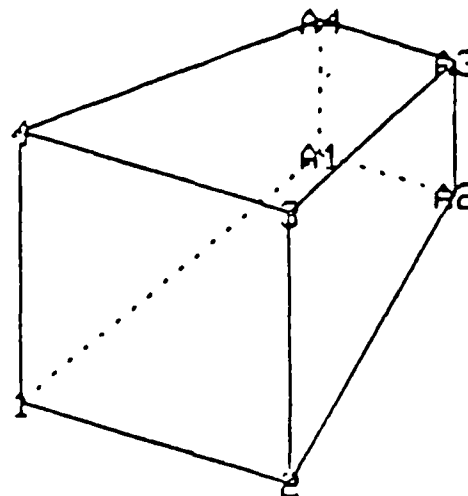
Figure 16. Scaling in Z only

Figure 17. Scaling in X and Z with nonzero apex, straight line segments

```

10 POIN 4
20 1 0. 0. 0.
50 2 10. 0. 0.
40 3 10. 0. 10.
30 4 0. 0. 10.
60 BLOCK BLI .1 15.
70 .5 .5 5. 5.
80 4 4 3 2 1

```



```

10 POIN 4
20 1 0. 0. -5.
30 2 5. 0. 0.
40 3 0. 0. 5.
50 4 -5. 0. 0.
60 CIRC 2 1 5.
70 CIRC 3 2 5.
80 CIRC 4 3 5.
90 CIRC 1 4 5.
100 BLOCK BLI .1 15.
110 .5 .5 0. 0.
120 4 4 3 2 1

```

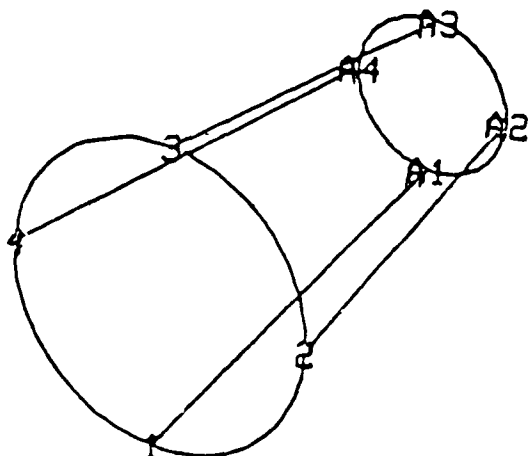


Figure 18. Scaling in X and Z with non-zero apex, circular line segments

```

10 POIN 4
20 1 0. 0. 0.
50 2 10. 0. 0.
40 3 10. 0. 10.
30 4 0. 0. 10.
60 BLOCK BLI .1 15.
70 .3 .3 5. 5. .8 .8
80 4 4 3 2 1

```

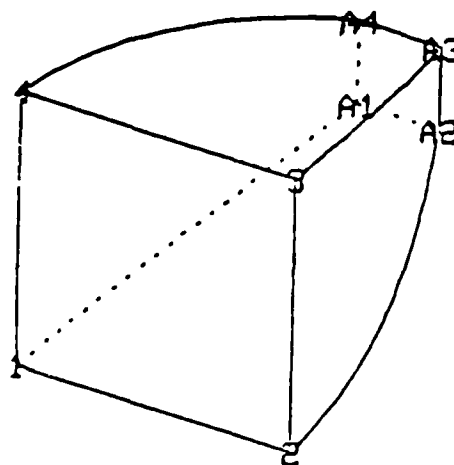


Figure 19. Quadratic variation

front face for all the connecting line segments. The (X, Y, Z) coordinates of each interpolation point are computed by

$$YINT = YOLD + DEPTH * .5$$

$$XINT = (XOLD - XAPEX) * HFX + XAPEX$$

$$ZINT = (ZOLD - ZAPEX) * HFZ + ZAPEX$$

"Face"

36. The "Face" command is used to define a solid object by giving a boundary representation of its faces. Faces can consist of

- a. Planar faces with straight or cubic sides.
- b. Bicubic patches defined by 16 (X, Y, Z) points.

The format for this command is as follows:

```
"Face" NAME DENS NFACE
"Face" DIR NAME DENS NFACE
"P" NPT PT1 PT2 . . . PTN
(WHERE N = NPT) (FOR PLANAR FACE)
    or
NPT PT1 PT2 . . . PTN
(WHERE N = NPT) (FOR PLANAR FACE)
    or
"C" PT1 PT2 . . . PT16 (FOR BICUBIC PATCH)
NOTE -- THE ABOVE INDENTED LINES ARE REPEATED
FOR EACH REMAINING FACE.
```

where NAME is a four character name, DENS is the density of the object being defined, and NFACE is the number of faces to be defined. Notice that next comes the connectivity of each face with a P to designate a planar face, and a C to designate a bicubic patch. If no letter is specified, P is the assumed default. The connectivity of a planar face consists of the number of points of the polygon followed by the point numbers. The order of the points must be counterclockwise if the outward normal to the face points out of the picture (-Y direction). The connectivity of a bicubic patch consists of the 16 points given a row at a time from left to right and from bottom to top (assuming the outward normal of the patch is pointing toward the observer). Figure 20 shows an eighth of a sphere defined using the 'Face' definition capability. The sample problem therefore consists of one bicubic patch and three planar faces. As before, the faces must be numbered so that a right-hand screw advances toward the outward normal (holes can be produced by reversing this process) when turned in the direction of the numbering.

37. Bicubic Patch. This patch was chosen not only for its ability to

```

10 RTZ 50. 50. 50. 50. 50. 51.
20 POIN 14
30 1 10. 0. 0.
40 2 10. 30. 0.
50 3 10. 60. 0.
60 4 10. 90. 0.
70 5 8.66 0. 5.
80 6 8.66 30. 5.
90 7 8.66 60. 5.
100 8 8.66 90. 5.
110 9 5. 0. 8.66
120 10 5. 30. 8.66
130 11 5. 60. 8.66
140 12 5. 90. 8.66
150 13 0. 0. 10.
160 14 0. 0. 0.
170 FACE SPHI .1 4
180 C 1 2 3 4 5 6 7 8 9 10 11 12 13 13 13 13
190 P 3 14 4 1
200 P 3 14 1 13
210 P 3 14 13 4

```

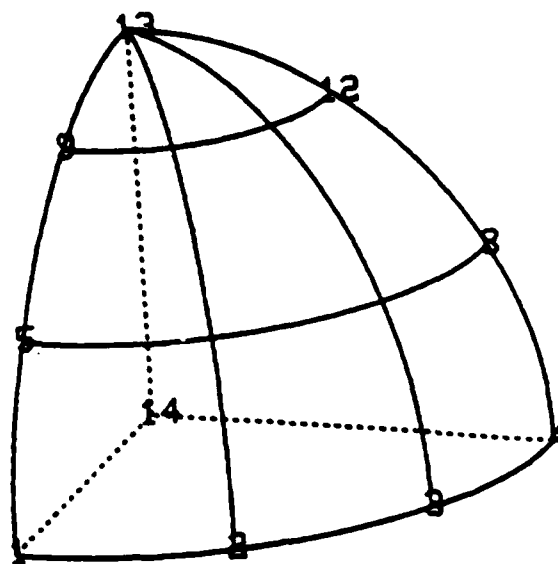


Figure 20. One-eighth of a sphere

model general shapes, but also for its accuracy when modeling conic sections. Note that it has 14 points given by rows as shown. Four points can also coincide (as was needed in the sample) to form a triangular shaped patch. If the patch is being used to model cones, cylinders, or spheres, it is important to place the points at equal angles apart. This rule was observed in the sample problem, and the volume is in error by 0.36 percent which should be quite satisfactory. Care must be taken in this patch to observe the right-hand screw rule.

38. Planar face. In an earlier version of the program, only straight-sided polygons were allowed. The cubic edges created from using bicubic patches can now be a part of a planar face. Note carefully however that in giving the connectivity data the two intermediate points of a cubic line segment are not given.

"BR8"

39. Another way to describe geometry is by the use of the eight-node brick element. Its format is

"BR8" NAME DENS

where NAME is a four-character name and DENS is the density. Following the command, the user gives the eight nodes to define the element and connectivity. Care must be taken to number the nodes of the element, insuring that they produce a positive volume. Figure 21 and the data file with it illustrate a typical example.

```

210 POIN 8
220 29 0 0 10
230 30 10 0 10
240 31 10 10 10
250 32 0 10 10
260 33 0 0 20
270 34 10 0 20
280 35 10 10 20
290 36 0 10 20
300 BRICK B1 100
310 29 30 31 32 33 34 35 36

```

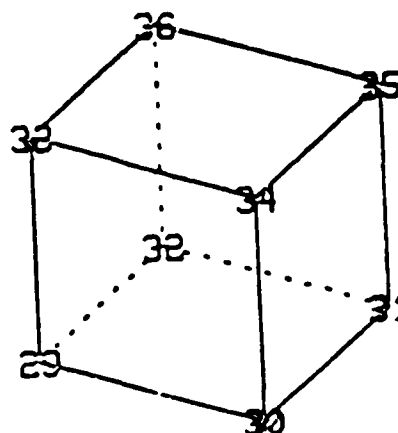


Figure 21. "BR8" command

"Translate"

40. Objects that have been defined can be moved or translated in space from one position to another using the "Translate" command. The possible options of this command are

```

"Translate"
"Translate" "POINTS"
"Translate" "POINTS" N1 N2
"Translate" "LINES"
"Translate" "LINES" N1 N2
"Translate" "DENSITY" DENS
"Translate" NAME

```

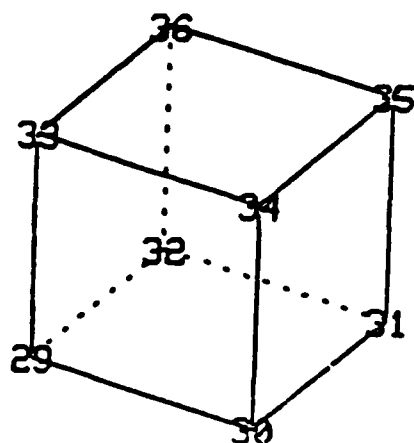
Note that points, lines, items with the same density, and items with the same name can be translated. Figure 22 shows an example of moving points by typing the following question and answer sequence:

```

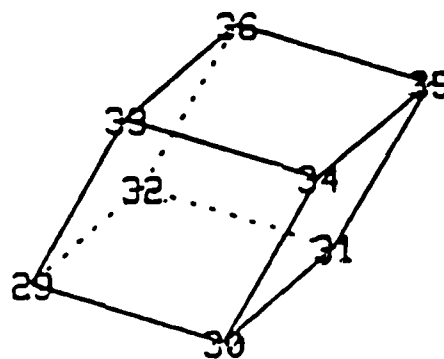
COMMAND ?
?TRAN POIN 29 32
INPUT DX, DY, DZ.
? -5., 0., 0.
COMMAND ?
?

```

Here DX, DY, and DZ are the increments by which the data are to be translated.



a. Before



b. After

Figure 22. "Translate" command

"ROTD"

41. This command allows the user to permanently rotate a piece of geometry about a specified axis. This is different from the "ROtp" command which only temporarily rotates the picture for easier viewing. The specific format of the command is

```
"ROTD" "H" ANGLE NAME XO YO ZO
"ROTD" "V" ANGLE NAME XO YO ZO
"ROTD" "O" ANGLE NAME XO YO ZO
```

H, V, and O are a horizontal, vertical, and outward axis, respectively, around which rotations are made. ANGLE is a counterclockwise positive angle in degrees that determines how much the piece is to be rotated about the given axis, NAME is the name of the piece that is to be rotated, and XO, YO, or ZO is a point that the given axis must go through.

42. The data file and resulting plots shown in Figure 23 illustrate the 'ROTD' command.

"COpy"

43. This command copies or duplicates the geometry of a given name and puts it into a new name. The format of this command is

```
"COpy NAME1 NAME2
```

where NAME1 is the geometry to be copied, and NAME2 is the name of the resulting geometry. NAME2 can now be translated or rotated in space. Figure 23 also illustrates the "COpy" command.

```

10 POIN 3
20 1 0 0 0
30 2 10 0 0
40 3 5 0 5
41 CIRC 2 1 7.072
50 BLOCK BL1 .1 2
60 1 1
70 3 3 2 1
71 LABEL GEOM
72 ISO
74 PLOT
80 COPY BL1 BL2
90 ROTD 0 90. BL2 5. 0. 5.
100 PLOT

```

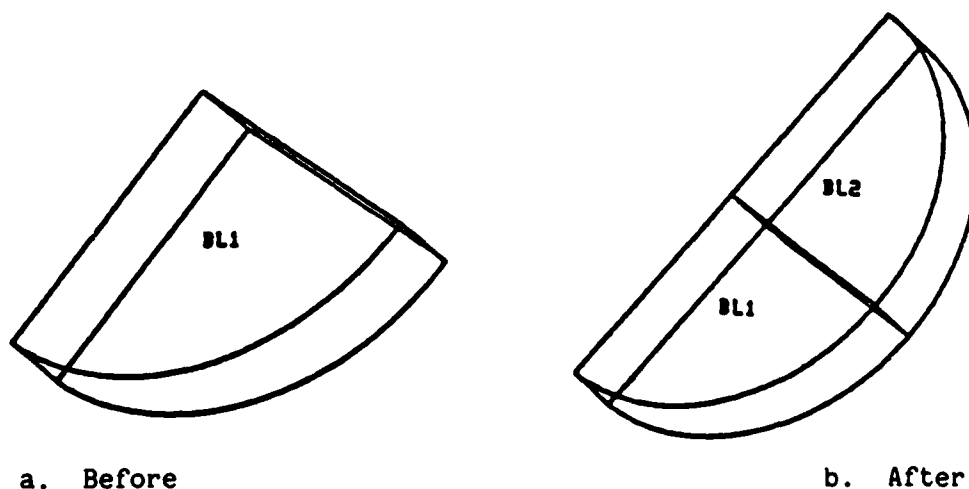


Figure 23. Example of "ROTD" command

"REFlect"

44. The REflect command allows the user to reflect or mirror a piece of geometry about a plane parallel to the X-Y, X-Z, or Y-Z planes. The format for this command is

```

"REFlect" "X" XVAL NAME
"REFlect" "Y" YVAL NAME
"REFlect" "Z" ZVAL NAME

```

For example, to reflect an object identified by "BL1" about the horizontal plane (parallel to the X-Y plane),

Z = 100

use the command

REF Z 100 BL1

Similar commands would be used for reflection about planes $X = XVAL$ and $Y = YVAL$. If "ALL" is given for NAME, the entire data base of geometry is reflected.

45. The data file and resulting plots shown in Figure 24 further illustrate the use of the REFlect command.

```
10 POINTS 7
20 1 80.436 0. 740.000
30 2 88.250 0. 896.270
40 3 100.000 0. 901.500
50 4 169.370 0. 847.930
60 5 212.966 0. 785.650
70 6 244.976 0. 756.344
80 7 244.976 0. 740.000
90 ELLI 3 2 11.750 5.230 RIGH
100 QUAD 3 4 134.685 0. 886.640
110 CIRC 6 5 100.000 LEFT
120 BLOCK OVE1 0.15500 49.000
130 1.1
140 7 1 2 3 4 5 6 7
150 COPY OVE1 OVE2
160 REFLECT X 0. OVE2
```

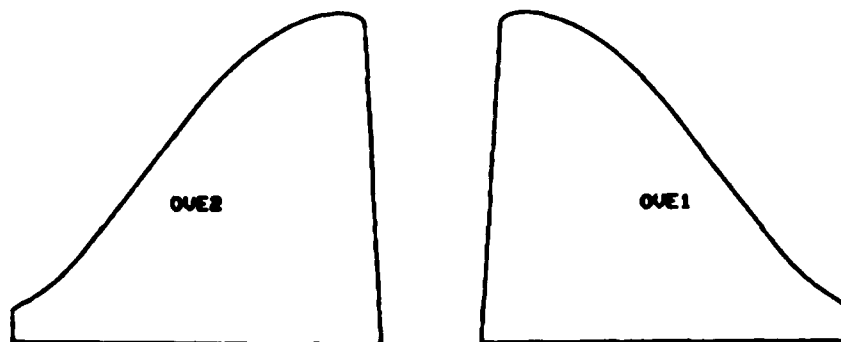


Figure 24. REFlect command example

"INPut"

46. This command allows the user to input or read into memory a permanent data file saved on disc. Its format is

```
"INPut" FLNM1
"INPut" FLNM1 "p"
```

where FLNM1 is a file description (20 characters maximum on Honeywell and 7 on CDC). If the P is also typed, a detailed printout of the input file is also typed, a detailed printout of the input file is printed on the terminal just as if it had been done interactively.

"Volume"

47. This command allows the user to obtain volume information whenever it is typed. Its format is

"Volume"
"Volume" "ALL"
"Volume" NAME

If "Volume" is typed, only the totals for the entire structure are given. If "Volume" NAME is provided, only the volume data for data having the name are given. "Volume" "ALL" will yield a detailed listing of the data by name; for example,

NO.	NAME	VOLUME	WEIGHT	XCG	YCG	ZCG
1	BL1	3,141.59	314,159.27	0.	20.00	0.
2	BL2	3,141.59	314,159.27	0.	20.00	25.00
	TOTAL	6,383.19	628,318.53	0.	20.00	12.50

"ENd"

48. This command is given to terminate running of the program. Its format is

"ENd"

"GO"

49. This command is used when the program is being used for a specific structure such as a dam or lock. Giving the "GO" command automatically caused the program to go on from the General Geometry Module to the General Loads Module. The format for this command is

"GO"

"REturn"

50. This command is used to return to the question

STRUCTURE TYPE OF GENERAL MODULE?

so that the user can select another module. Typically, the next module is the General Loads Module. The format for this command is

"REturn"

"CLip"

51. The "CLip" command allows the user to cut or clip the geometry by an arbitrary plane to produce a new structure for use as a free-body diagram or making new weight and centroid computations. The clipped data will be placed into a new data file specified by the user. Blocks, bricks, or faces not touched by the clipping plane will be placed untouched in the new file. The parts left over from a clip however will be converted to faces with the curved parts being modeled by bicubic patches and planar pieces having both straight and cubic sides. The specific format of the command is

```
"CLip X" XVAL NAME
"CLip +X" XVAL NAME
"CLip -X" XVAL NAME
"CLip Y" YVAL NAME
"CLip +Y" YVAL NAME
"CLip -Y" YVAL NAME
"CLip Z" ZVAL NAME
"CLip +Z" ZVAL NAME
"CLip -Z" ZVAL NAME
"CLip" LAB1 LAB2 LAB3 NAME
```

Note that you can clip parallel to the major axes or use three previously defined points in space to define a plane to do the clippings. "CLip X" XVAL means clip according to the vertical plane parallel to the YZ plane at a value of $X = XVAL$, "CLip Y" YVAL means clip according to the vertical plane parallel to the XZ plane at a value of $Y = YVAL$, and "CLip Z" ZVAL means clip according to the horizontal plane $Z = ZVAL$. A positive X, Y, or Z indicates that everything in the positive direction past the specified plane should be kept and the other clipped-off part thrown away, whereas a negative X, Y, or Z designation indicates keeping in the negative direction. LAB1, LAB2, and LAB3 are the labels of three previously defined points in space that determine the clipping plane. The part that is kept is pointed to as a right-hand screw and is advanced in the same way that the points are specified. NAME is optional and specifies a specific piece of structure to be clipped. All other pieces of geometry are left untouched and placed into the output file. If NAME is not specified, the entire geometry is clipped.

52. When the "CLip" command is given, the program asks for two output data files where the new clipped data are to be placed. The first data file contains the solid geometry description (blocks, bricks, faces, etc.), and the

second contains a definition of the resulting base or cross section of the intersection of the clipping plane and the original geometry. After the clip is complete, the program returns to the command level.

53. A sample data file (called SAMDAT) and the resulting interaction after giving a "CLIP" command are shown in Figure 25 with before and after clips plots.

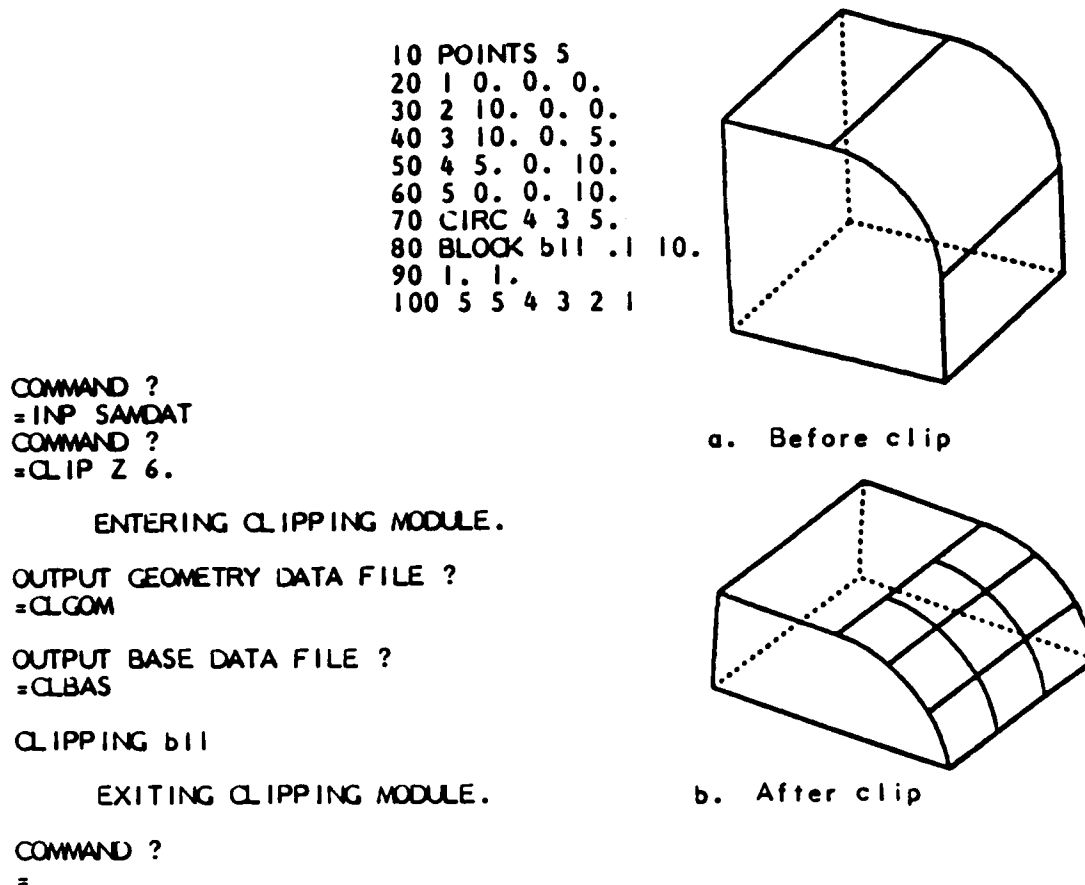


Figure 25. "CLIP" command example

"CLear"

54. This command is used to clear all definition of geometry from memory to begin a new problem. Its format is

"CLear"

"PLot"

55. This command allows part or all of the data base to be plotted. Its format is

```

"PLot"
"PLot" "POINTs"
"PLot" "POINTs" N1 N2
"PLot" "LINEs"
"PLot" "LINEs" N1 N2
"PLot" "DENSity" DENS
"PLot" NAME

```

By typing "PLot", all the lines will be plotted. Thus, the default code is "LINEs". By giving N1 and N2, a portion of the total number of lines can be plotted. Points can only be plotted by typing "PLot" "POINTs". If N1 and N2 are given, only those points between N1 and N2 are plotted. Also, everything with a given density (DENS) can be plotted. All data with a name (NAME) can be plotted as well.

"Window"

56. This command allows the user to pick a window, or portion of the plot on the screen, and plot just that portion. Its format is

```
"Window"
```

After typing "Window", the cross hairs will appear. Place them on the lower left-hand corner of the desired window, and type any character and carriage return (some graphics terminals are strapped so the carriage return is not needed). The cross hairs will reappear and the process will be repeated, except that this time place the cross hairs on the upper right-hand corner of the window. Figure 26 shows a full picture of a brick element, while Figure 27 shows a window of that plot.

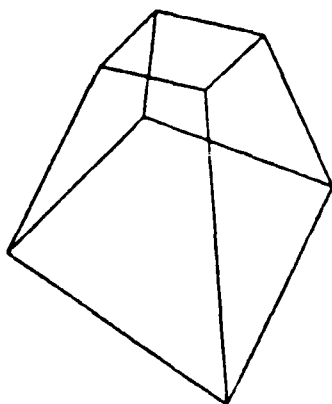


Figure 26. Full picture
of brick element



Figure 27. Window
of Figure 26 brick
element

"Zoom"

57. This command allows the user to decrease or increase the size of the current picture on the screen. The format for this command is

"Zoom" FMAG

where FMAG is a scale factor which dictates whether the current picture is made bigger or smaller. If FMAG is less than one, the picture is decreased in size; if FMAG is greater than one, the picture is increased in size.

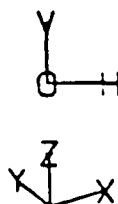
"ROtp"

58. This command allows the user to rotate a picture of the structure for viewing at different angles. This command is different from the "ROTD" command which permanently rotates the geometry to a new position. The ROTP command changes only the picture on the screen while the permanent storage of the geometry is left untouched. Its format is

"ROtp" "H" ANGLE
"ROtp" "V" ANGLE
"ROtp" "O" ANGLE

where H means horizontal axis, V means vertical axis, and O means outward axis. ANGLE is a counterclockwise, positive angle in degrees that determines how much the structure is to be rotated about the given axis. Note, as shown in the following example, the distinction between the coordinate system of the X, Y, and Z data and that of that of the rotation axes. A good set of rotation is

"ROtp" "V" 30.
"ROtp" "H" 30.



"ISometric"

59. This command allows the user to specify a standard set of rotations rather than search for the desired plot. Its format is

"ISometric"

It is equivalent to the two rotations given above.

"Label"

60. This command allows labels to be plotted along with line

segments. Two types of labels are available: (1) point labels and (2) name labels. The format for this command is

```
"Label" "Points"  
"Label" "Geometry"
```

Figure 28 illustrates the labeling of points by showing the results of typing

```
LABEL POINTS  
PLOT
```

Figures 23 and 24 show examples of labeling the names of geometric pieces.

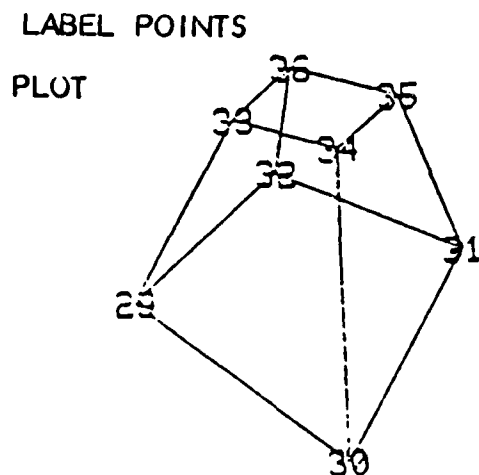


Figure 28. "Label" "Points"
command

"Nolabel"

61. This command turns off the label options of the "Label" command. Its format is

```
"Nolabel" "Points"  
"Nolabel" "Geometry"
```

"Nolabel" is the default over "Label".

"Dash"

62. This command allows the user to dash the lines that are hidden from view. The format for this command is

```
"Dash"
```

Figure 29 shows the results of dashing the plot in Figure 26.

"Hide"

63. The command allows the user to delete hidden lines from the plot.

Its format is

"Hide"

Figure 30 shows the result of using the HIDE command on the plot in Figure 26.

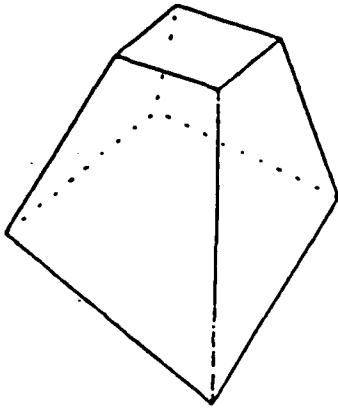


Figure 29. "Dash" command

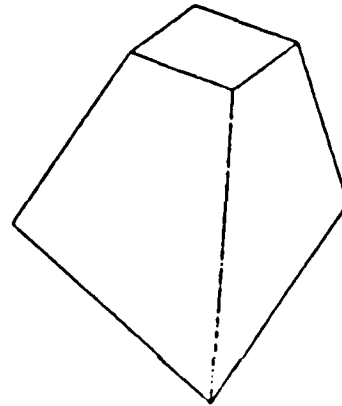


Figure 30. "Hide" command

"Solid"

64. This command causes all the lines to be solid rather than dashed or deleted. The format for this command is

"Solid"

"Solid" is default over "Dash" and "Hide".

"ERase"

65. This command erases the screen. This is needed where there is no built-in erase mechanism for the system. Its format is

"ERase"

"INitalize"

66. This command initializes plot data back to the original values. Its format is

"INItialize"

Currently, the angles of rotation are reset to zero.

Practical Applications

67. The applicability of this program can be studied in further detail with the plates herein. Additional study and research have been done by Mr. Obbie Thompson, Jr.,* US Army Engineer District, St. Louis, though the material has not been published. Plates 1 and 2 are plots provided by Messrs. Thomas J. Mudd* and John Jobst,* US Army Engineer District, St. Louis, illustrating practical applications of this program. Plate 3 was provided by Mr. Steve Freitas,** Sacramento District. These plates are placed at the end of the main text of this report.

* The contributions by Messrs Thompson, Mudd, and Jobst, US Army Engineer District, St. Louis, are the result of preliminary work in preparation for Design Memorandum No. 17, 1978 (Dec), John H. Overton Lock and Dam, Red River Waterway.

** The plate furnished by Mr. Steve Freitas, US Army Engineer District, Sacramento, has never been published.

REFERENCES

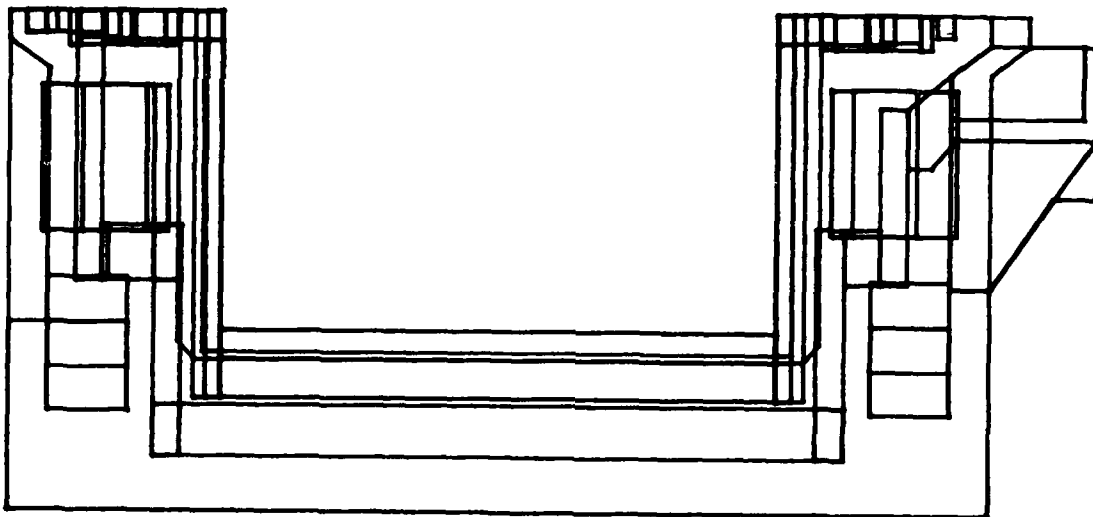
Johnson, Robert H. 1984. Solid Modeling: A State-of-the-Art Report, CAD/CAM ALERT, Management Roundtable, Inc., Chestnut Hill, Mass.

Tracy, Fred T. 1980. "A Three-Dimensional Analysis/Design Program (3DSAD), General Geometry Module," Instruction Report K-80-4, Report 1, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

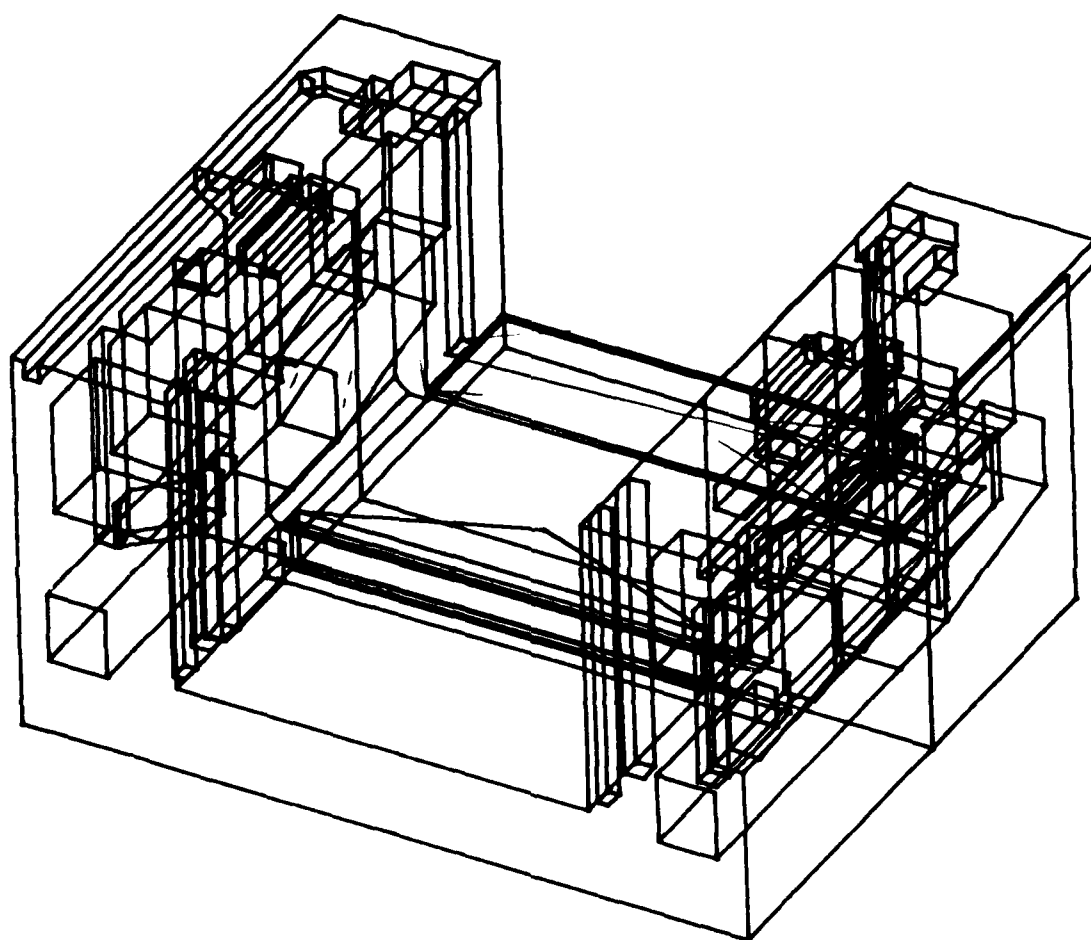
Tracy, Fred T., and Kling, Charles W. 1982. "A Three-Dimensional Analysis/Design Program (3DSAD), General Analysis Module," Instruction Report K-80-4, Report 3, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Tracy, Fred T., Kling, Charles W., and Holtham, William J. 1983. "A Three-Dimensional Analysis/Design Program (3DSAD), Special Purpose Modules for Dams (CDAMS)," Instruction Report K-80-4, Report 4, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

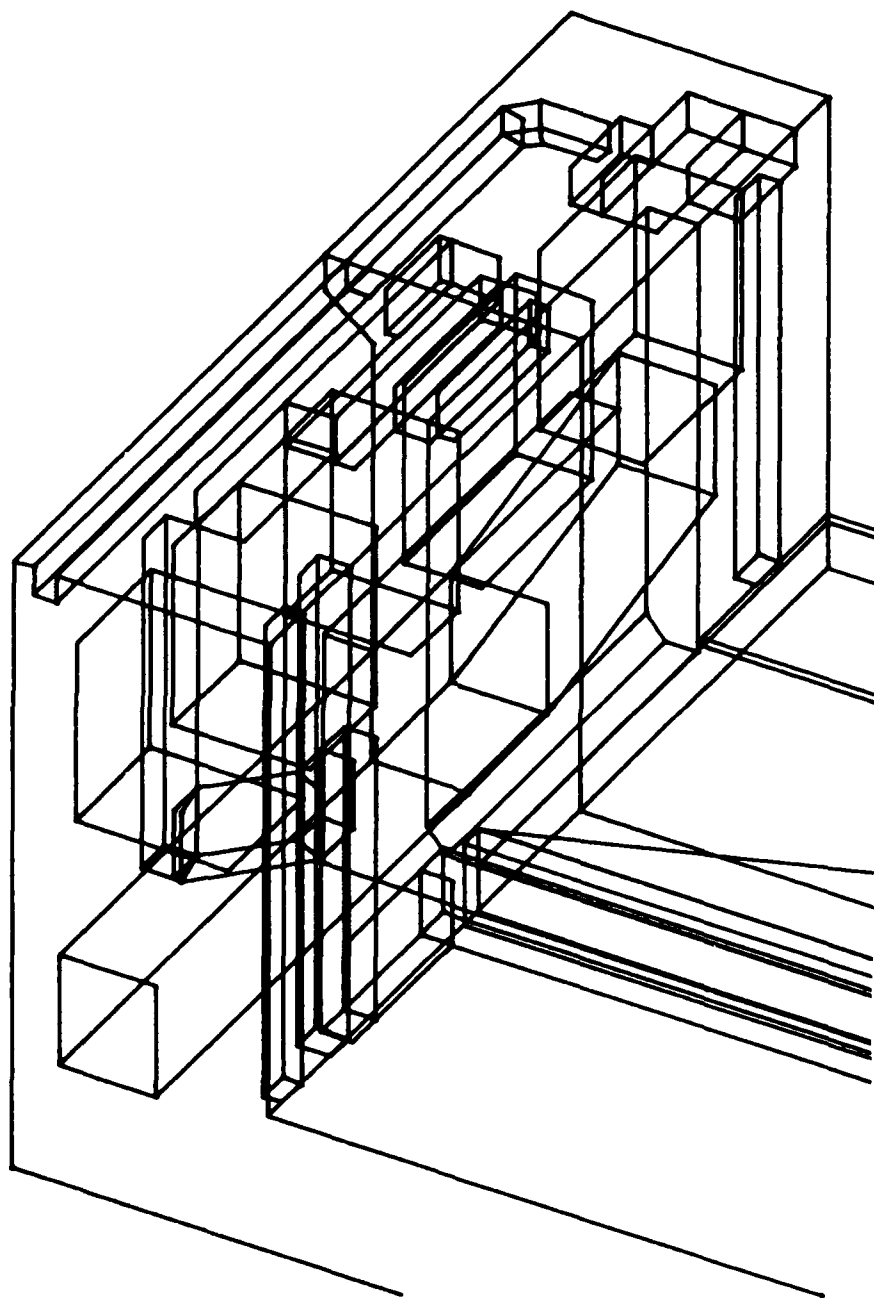
Wilson, Howard B., Jr., and Farrior, Donna S. 1976. "Computation of Geometric and Inertial Properties for General Areas and Volumes of Revolution," Computer Aided Design, Vol 8, No. 4.



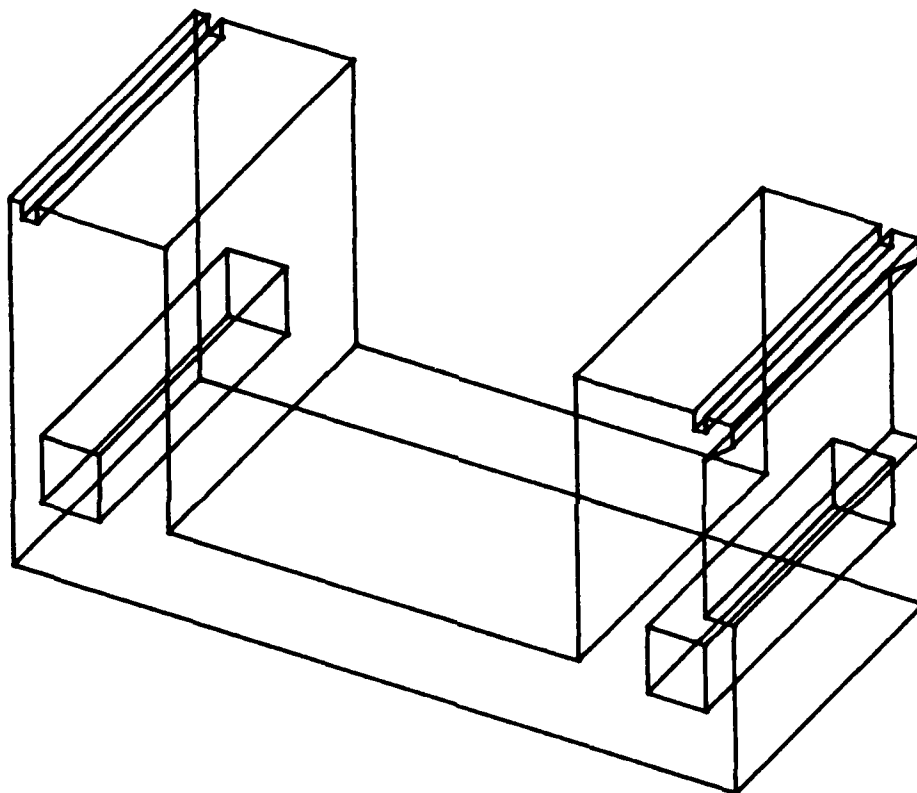
STRUCTURE 1
Front View



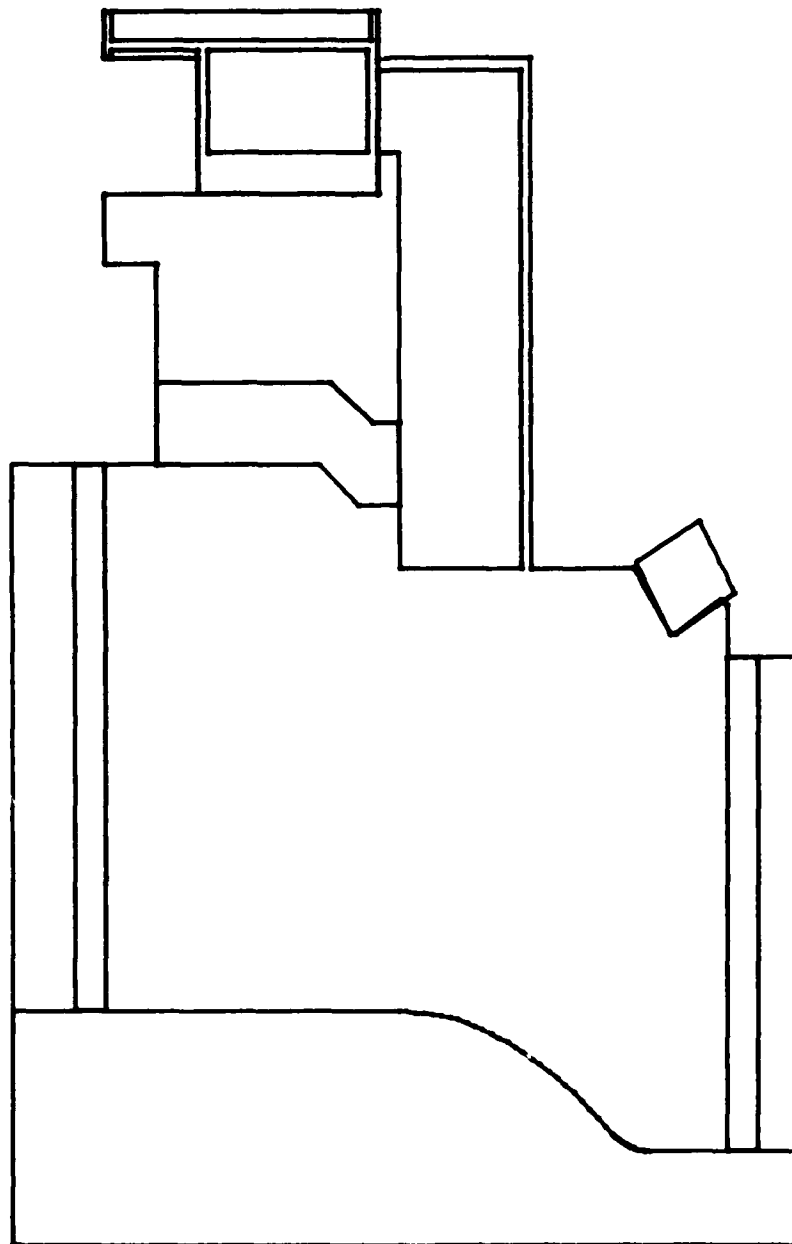
STRUCTURE 1
Overhead View



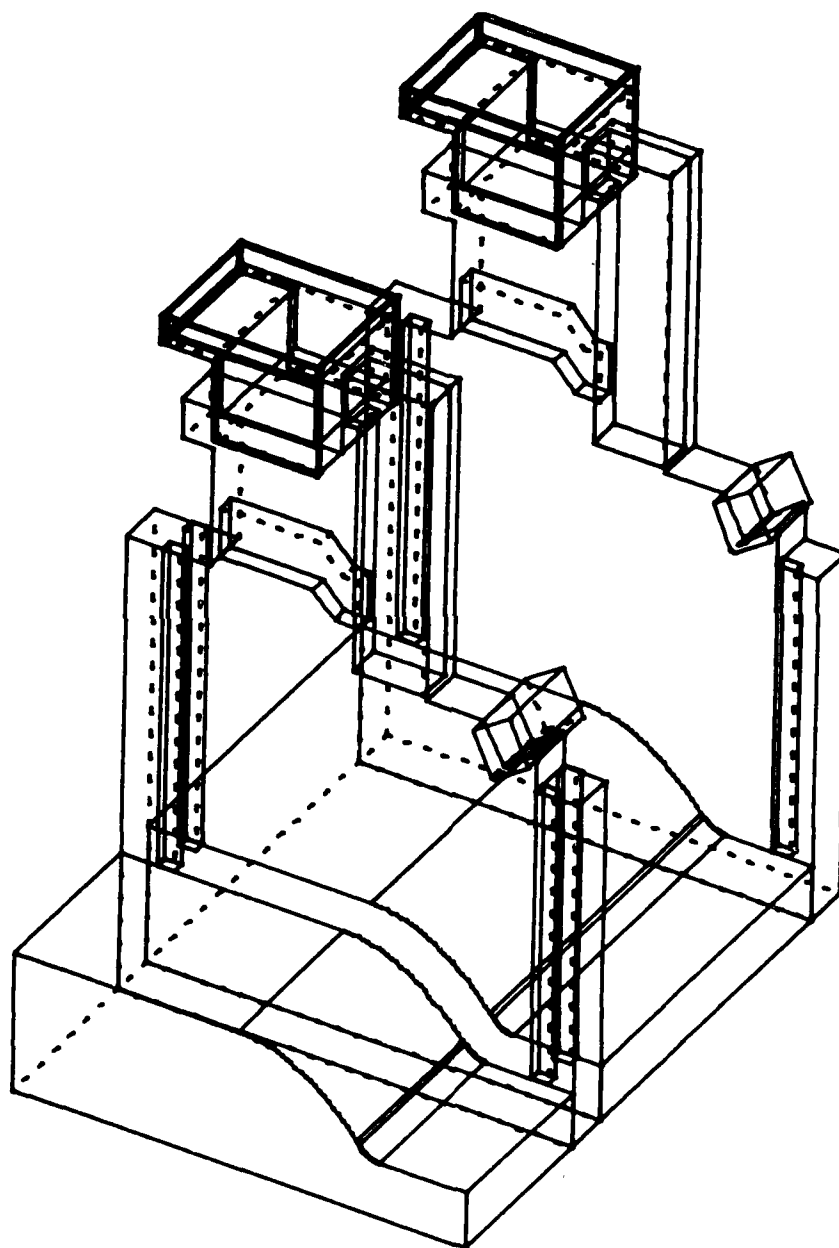
STRUCTURE 1
Window of
Overhead View



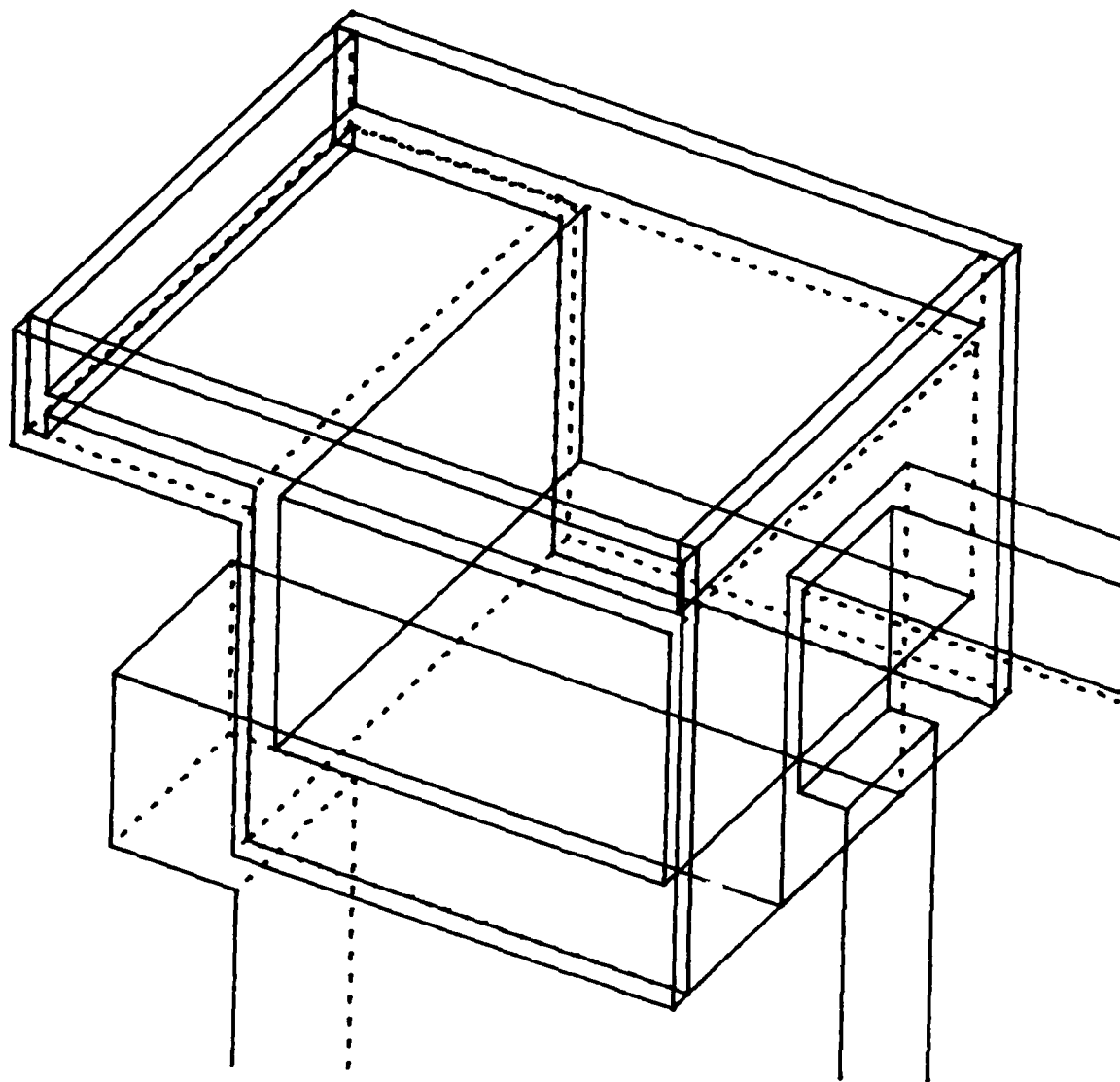
STRUCTURE 1
First Block



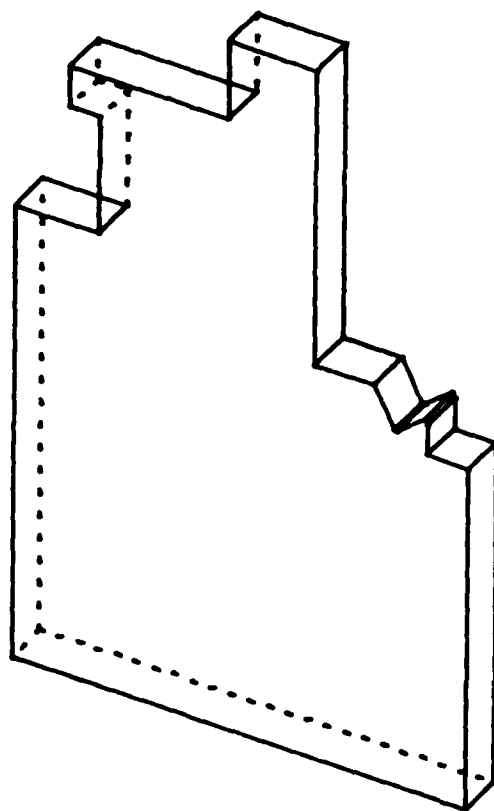
STRUCTURE 2
Front View



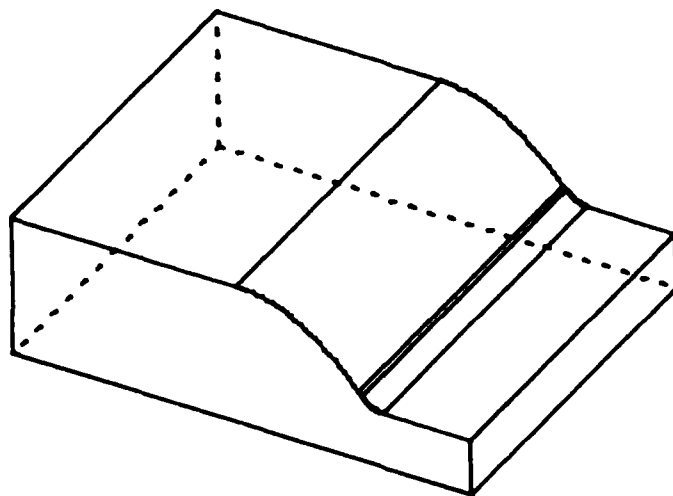
STRUCTURE 2
Overhead View



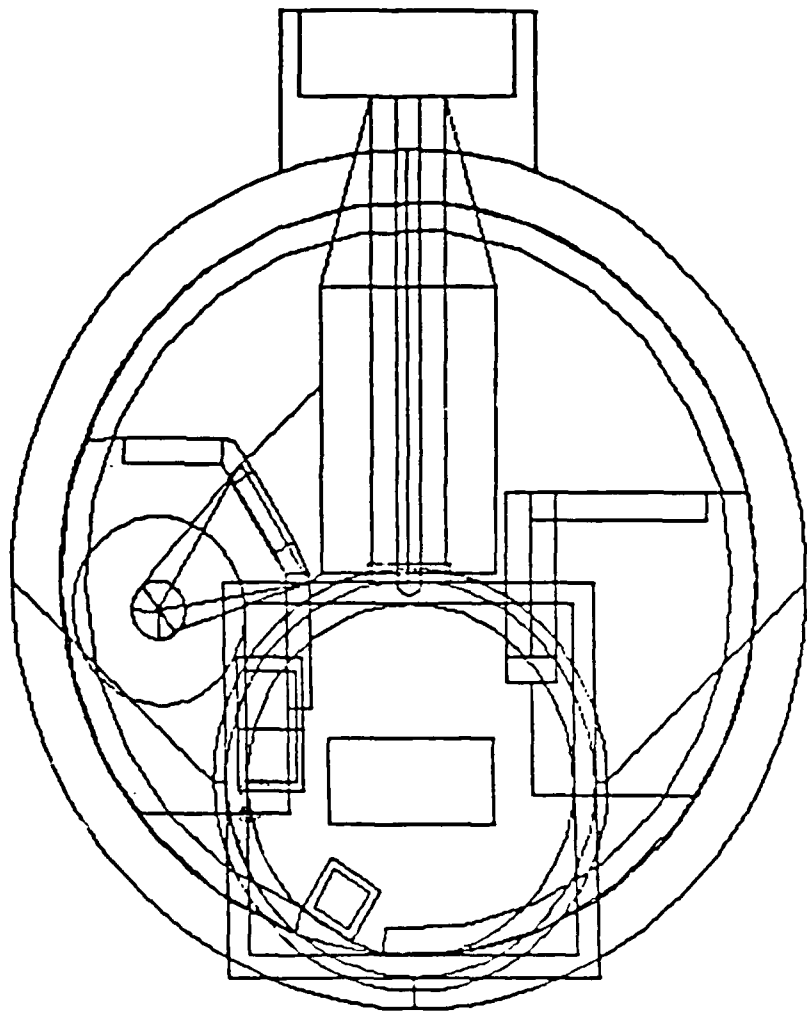
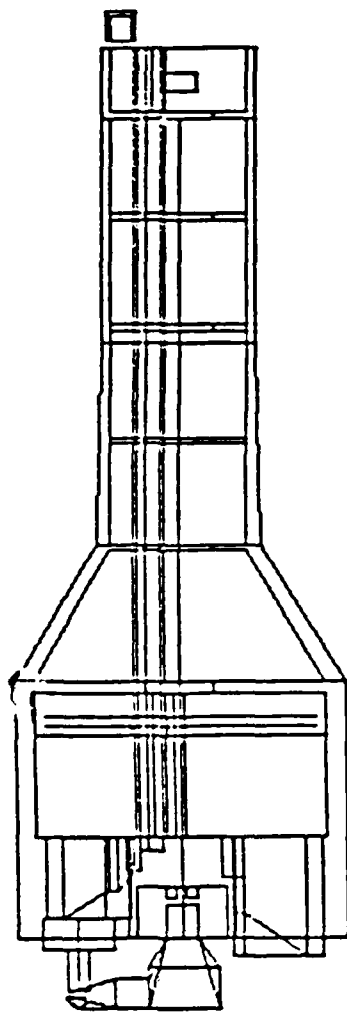
STRUCTURE 2
Window of
Overhead View



STRUCTURE 2
First Block



STRUCTURE 2
Second Block



STRUCTURE 3

APPENDIX A: THEORY AND PROCEDURE FOR COMPUTING MASS PROPERTIES

Introduction

1. A Three-Dimensional Stability Analysis/Design (3DSAD) Program (Tracy 1980;* Tracy and Kling 1982; Tracy, Kling, and Holtham 1983) for analyzing and designing concrete structures has been developed and successfully used. This program models the geometry and loads in a general way and then applies this modeling capability to specific structures types whenever possible. Examples include dams with overflow, nonoverflow, and pier sections; gravity and U-frame locks; cooling towers; etc. Even if the structure does not conform to a predefined standard shape, the program can do a general analysis of the problem because of the fundamental approach of using solid modeling type capability as a foundation.

2. There are several ways to model geometry and many commercial packages exist for this (Johnson 1984). The approach taken in 3DSAD was to first provide some of this capability internal to the program and at a later date supply the capability to tie 3DSAD into the larger system.

Modeling Technique

3. 3DSAD models geometry in three ways:

- a. Blocks--Two-dimensional (2-D) cross sections swept normal to the cross section in a linear or axisymmetric way with linear and quadratic tapering for the linear sweeps.
- b. Eight-node brick elements.
- c. A boundary representation consisting of planar faces and bicubic patches.

Mass Properties--BLOCK

4. One of the major points of this appendix is to describe the

* References cited in this appendix are included in the References at the end of the main text.

equations for performing mass properties for blocks. The blocks are formulated by first defining a cross section, then parametrically performing a constant or tapered linear sweep or an axisymmetric sweep. Thus, area integrals can first be computed, and from these the mass properties can be figured by performing the integration in the third dimension.

5. For a cross section defined by a polygon, it is best to convert the area integrals to line integrals (Wilson and Farrior 1976). For a cross section consisting of both curved and straight line segments, it is tempting to first approximate the curved edges with straight line segments and then the line integrals. However, this process is time consuming and creates unnecessary errors, since it is possible to formulate line integrals for curved edges as well. This appendix presents examples of these line integrals developed in a consistent manner.

6. The volume of a block with constant cross section (Figure A1) having j sides can be expressed as

$$\begin{aligned} V &= -L \int_C Y dX \\ &= -L \sum_{j=1}^n \int_{C_j} Y dX \\ &= -L \sum_{j=1}^n I_j \end{aligned}$$

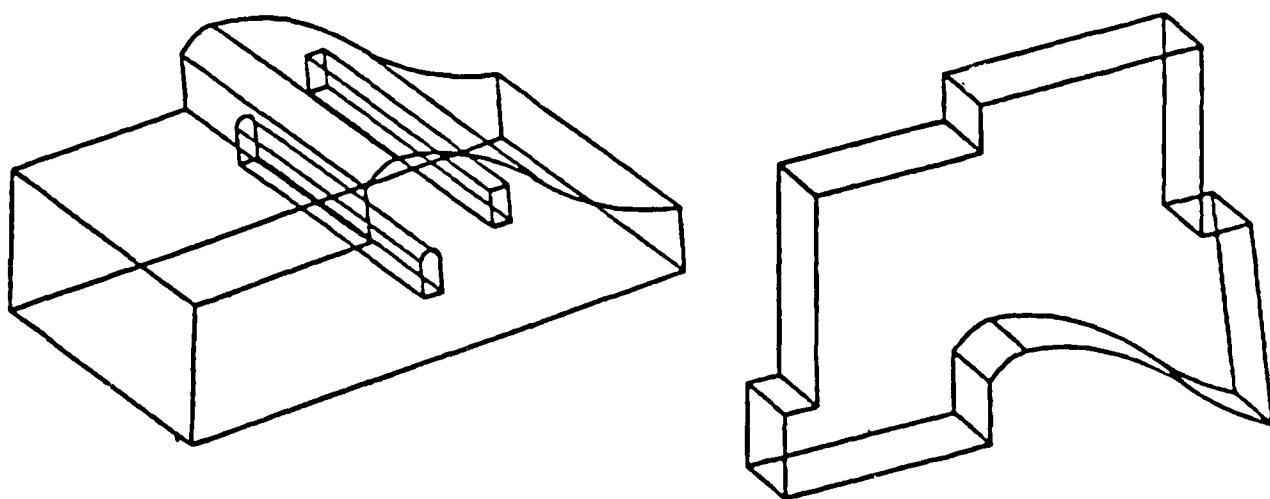


Figure A1. Blocks with constant cross section

The integral I_j can now be evaluated for different line segments types.

7. Linear line segment. Suppose the line segment is defined by two points $(X_1, Y_1) - (X_2, Y_2)$. Then

$$X = X_1 + (X_2 - X_1)S$$

$$Y = Y_1 + (Y_2 - Y_1)S$$

where S varies between 0 and 1. I can be done as:

$$\begin{aligned} I &= \int_0^1 [Y_1 + (Y_2 - Y_1)S] (X_2 - X_1) dS \\ &= \int_0^1 A_1 \sum_{k=0}^1 B_k S^k dS \\ &= A_1 \sum_{k=0}^1 \frac{B_k}{k+1} \end{aligned}$$

8. Quadratic line segment. In a similar manner, a quadratic line segment formed by three points, 1, 2, and 3, can be expressed as:

$$X = [X_1 \ X_2 \ X_3] M \begin{bmatrix} 1 \\ S \\ S^2 \end{bmatrix}$$

$$= \sum_{k=0}^2 A_k S^k$$

$$Y = \sum_{m=0}^2 B_m S^m$$

$$dX = \sum_{k=0}^1 (k+1)A_{k+1} S^k dS$$

Here M is the "magic" matrix of known constants, and as in the linear case, A and B are constants of the parametric polynomials. The line integral I can now be easily evaluated as follows:

$$I = \int_0^1 \left(\sum_{m=0}^2 B_m S^m \right) \left[\sum_{k=0}^1 (k+1) A_{k+1} S^k \right] dS$$

$$= \sum_{m=0}^2 \sum_{k=0}^1 \frac{k+1}{k+m+1} A_{k+1} B_m$$

Note also at this point that higher order integrals, the X centroid, for example, can be done with equal ease.

$$\bar{X} = -\frac{L}{V} \int_C XY dX$$

$$I = \sum_{m=0}^2 \sum_{n=0}^2 \sum_{k=0}^1 \frac{k+1}{k+m+n+1} A_m B_n A_{k+1}$$

The general trend is now set with the form being the same for any polynomial. Also, another X or Y inside the integral simply results in another summation sign.

9. Circular arc. The circular arc can also be put in the same consistent notation with use of complex variables as:

$$X = R \cos (AS + \theta_0)$$

$$Y = R \sin (AS + \theta_0)$$

$$W = e^{iAS}$$

$$X = \sum_{k=-1}^1 A_k W^k$$

$$Y = \sum_{m=-1}^1 B_m W^m$$

$$dX = iA \sum_{k=-1}^1 k A_k W^k dS$$

With this foundation the line integrals can be figured as easily as the polynomials.

$$I = \int_0^1 \left(\sum_{m=-1}^1 B_m w^m \right) \left(iA \sum_{k=-1}^1 k A_k w^k \right) dS$$

$$= \sum_{m=-1}^1 \sum_{k=-1}^1 \frac{k}{k+m} \left[e^{iA(k+m)} - 1 \right]$$

10. Elliptical arc. An elliptical arc can also be handled in the same consistent manner.

$$X = A \cos (\theta S + \theta_0)$$

$$Y = B \sin (\theta S + \theta_0)$$

$$X = \sum_{k=-1}^1 A_k w^k$$

$$Y = \sum_{m=-1}^1 B_m w^m$$

11. Tapered block. Consider now a cross section defined in the X-Y plane. Let (X_0, Y_0) be a point on that cross section. The (X, Y) value of a point on a cross section at elevation Z as a result of a taper is

$$X(Z) = (X_0 - X_A) S_X(Z) + X_A$$

$$Y(Z) = (Y_0 - Y_A) S_Y(Z) + Y_A$$

where (X_A, Y_A) is an apex point, and $S_X(Z)$ and $S_Y(Z)$ are scale factors that range in value from 0 to 1. The scale factor functions determine the type of tapering. Let Z vary from 0 to L and let the scale factor functions vary quadratically:

$$Z = LS$$

$$S_X = \begin{bmatrix} 1 & H_X & F_X \end{bmatrix}^M \begin{bmatrix} 1 \\ S \\ S^2 \end{bmatrix}$$

$$= \sum_{j=0}^2 A_j S^j$$

$$S_Y = \sum_{k=0}^2 B_k S^k$$

where H_X and F_X are the values of the X scale factor, respectively, for $S = 0.5$ and $S = 1$. Let a similar expression for the Y scale factor also be defined. Let A be the area of the cross section at elevation Z. Then the volume is completed by

$$A = A_0 S_X S_Y$$

$$V = \int_0^L A dz$$

$$= A_0 \int_0^1 \left(\sum_{j=0}^2 A_j S^j \right) \left(\sum_{k=0}^2 B_k S^k \right) L dS$$

$$= A_0 L \sum_{j=0}^2 \sum_{k=0}^2 \frac{1}{j+k+1} A_j B_k$$

Mass Properties--FACES

12. Two types of faces will be considered:

- a. Planar face
- b. Bicubic patch

13. Planar face. By using the equation of the plane, the mass property integrals for planar faces can also be converted to line integrals and then computed as before. For example, the volume under a planar face can be computed as:

$$\begin{aligned}
V &= \int_A Z \, dX \, dY \\
&= \int_A (AX + BY + C) \, dX \, dY \\
&= - \oint_C (AX + .5BY' + C) \, YdX
\end{aligned}$$

These volumes, when summed over all the faces, will give the correct volume for the solid.

14. Bicubic patch. Bicubic patches can be defined in several ways. Whatever the boundary conditions or formulation, they can typically be cast:

$$\begin{aligned}
X &= \sum_{i=0}^3 \sum_{j=0}^3 A_{ij} S^i T^j \\
Y &= \sum_{k=0}^3 \sum_{l=0}^3 B_{kl} S^k T^l \\
Z &= \sum_{m=0}^3 \sum_{n=0}^3 C_{mn} S^m T^n
\end{aligned}$$

Mass properties can be computed with numerical integration or exact formulation. The exact formulation is:

$$\begin{aligned}
V &= \int_A Z \, dX \, dY \\
&= \int_0^1 \int_0^1 Z \, |J| \, dS \, dT \\
&= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 T_{ijklmn} \\
T_{ijklmn} &= \frac{(il - jk) A_{ij} B_{kl} C_{mn}}{(i + k + m) (j + l + n)}
\end{aligned}$$

WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

	Title	Date
Technical Report K-78-1	List of Computer Programs for Computer-Aided Structural Engineering	Feb. 1978
Instruction Report O-79-2	User's Guide. Computer Program with Interactive Graphics for Analysis of Plane Frame Structures (CFRAME)	Mar. 1979
Technical Report K-80-1	Survey of Bridge-Oriented Design Software	Jan. 1980
Technical Report K-80-2	Evaluation of Computer Programs for the Design/Analysis of Highway and Railway Bridges	Jan. 1980
Instruction Report K-80-1	User's Guide. Computer Program for Design/Review of Curvilinear Conduits/Culverts (CURCON)	Feb. 1980
Instruction Report K-80-3	A Three-Dimensional Finite Element Data Edit Program	Mar. 1980
Instruction Report K-80-4	A Three-Dimensional Stability Analysis/Design Program (3DSAD)	
	Report 1. General Geometry Module	Jun. 1980
	Report 3. General Analysis Module (CGAM)	Jun. 1980
	Report 4. Special-Purpose Modules for Dams (CDAMS)	Aug. 1980
Instruction Report K-80-6	Basic User's Guide. Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA)	Dec. 1980
Instruction Report K-80-7	User's Reference Manual. Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA)	Dec. 1980
Technical Report K-80-4	Documentation of Finite Element Analyses	
	Report 1. Longview Outlet Works Conduit	Dec. 1980
	Report 2. Anchored Wall Monolith, Bay Springs Lock	Dec. 1980
Technical Report K-80-5	Basic Pile Group Behavior	Dec. 1980
Instruction Report K-81-2	User's Guide. Computer Program for Design and Analysis of Sheet Pile Walls by Classical Methods (CSHTWAL)	
	Report 1. Computational Processes	Feb. 1981
	Report 2. Interactive Graphics Options	Mar. 1981
Instruction Report K-81-3	Validation Report. Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA)	Feb. 1981
Instruction Report K-81-4	User's Guide. Computer Program for Design and Analysis of Cast-in-Place Tunnel Linings (NEWTUN)	Mar. 1981
Instruction Report K-81-6	User's Guide. Computer Program for Optimum Nonlinear Dynamic Design of Reinforced Concrete Slabs Under Blast Loading (CBARCS)	May 1981
Instruction Report K-81-7	User's Guide. Computer Program for Design or Investigation of Orthogonal Culverts (CORTCUL)	Mar. 1981
Instruction Report K-81-9	User's Guide. Computer Program for Three-Dimensional Analysis of Building Systems (CTABBS80)	Apr. 1981
Technical Report K-81-2	Theoretical Basis for CTABBS80. A Computer Program for Three-Dimensional Analysis of Building Systems	Apr. 1981
Instruction Report K-81-7	User's Guide. Computer Program for Analysis of Reinforced Structures with Nonlinear Supports (CRFAM)	Apr. 1981
Instruction Report K-81-7	User's Guide. Computer Program for Design or Investigation of Reinforced Structures with Nonlinear Supports (CRFAM)	Apr. 1981

WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

(Concluded)

	Title	Date
Instruction Report K-83-1	User's Guide: Computer Program With Interactive Graphics for Analysis of Plane Frame Structures (CFRAME)	Jan 1983
Instruction Report K-83-2	User's Guide: Computer Program for Generation of Engineering Geometry (SKETCH)	Jun 1983
Instruction Report K-83-5	User's Guide: Computer Program to Calculate Shear, Moment, and Thrust (CSMT) from Stress Results of a Two-Dimensional Finite Element Analysis	Jul 1983
Technical Report K-83-1	Basic Pile Group Behavior	Sep 1983
Technical Report K-83-3	Reference Manual: Computer Graphics Program for Generation of Engineering Geometry (SKETCH)	Sep 1983
Technical Report K-83-4	Case Study of Six Major General-Purpose Finite Element Programs	Oct 1983
Instruction Report K-84-2	User's Guide: Computer Program for Optimum Dynamic Design of Nonlinear Metal Plates Under Blast Loading (CSDOOR)	Jan 1984
Instruction Report K-84-7	User's Guide: Computer Program for Determining Induced Stresses and Consolidation Settlements (CSETT)	Aug 1984
Instruction Report K-84-8	Seepage Analysis of Confined Flow Problems by the Method of Fragments (CFRAG)	Sep 1984
Instruction Report K-84-11	User's Guide for Computer Program CGFAG, Concrete General Flexure Analysis with Graphics	Sep 1984
Technical Report K-84-3	Computer-Aided Drafting and Design for Corps Structural Engineers	Oct 1984
Technical Report ATC-86-5	Decision Logic Table Formulation of ACI 318-77, Building Code Requirements for Reinforced Concrete for Automated Constraint Processing, Volumes I and II	Jun 1986
Technical Report ITL-87-2	A Case Committee Study of Finite Element Analysis of Concrete Flat Slabs	Jan 1987
Instruction Report ITL-87-1	User's Guide: Computer Program for Two-Dimensional Analysis of U-Frame Structures (CUFRAM)	Apr 1987
Instruction Report ITL-87-2	User's Guide: For Concrete Strength Investigation and Design (CASTR) in Accordance with ACI 318-83	May 1987
Technical Report ITL-87-6	Finite-Element Method Package for Solving Steady-State Seepage Problems	May 1987
Instruction Report ITL-87-3	User's Guide: A Three Dimensional Stability Analysis/Design Program (3DSAD), Report 1, Revision 1: General Geometry Module	Jun 1987

END

9-87

DTIC